Um estudo sobre os desafios de segurança na adoção da Arquitetura Orientada a Serviços.

Claudio Aparecido Rocha

Trabalho Final de
Mestrado Profissional

TERMO DE APROVAÇÃO

Trabalho Final Escrito defendido e aprovado em 24 de julho de 2006, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Antonio Montes Filho
CenPRA

Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP

Prof. Dr. Ricardo Dahab
IC - UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Rocha, Claudio Aparecido

Um estudo sobre os desafios de segurança na adoção da Arquitetura Orientada a Serviços (SOA) / Claudio Aparecido Rocha --- Campinas, [SP :], 2006

Orientador: Ricardo Dahab

Trabalho Final (Mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Segurança. 2. Arquitetura Orientada a Serviços 3. SOA. 4. XKMS. I. Dahab, Ricardo. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Um estudo sobre os desafios de segurança na adoção da Arquitetura Orientada a Serviços

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigido e defendido por Cláudio Aparecido Rocha e aprovado pela Banca Examinadora

Campinas, 31 de Maio de 2007.

Prof. Dr. Ricardo Dahab
(Orientador)

Trabalho final apresentado ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Computação na área de Engenharia da Computação.

“Uma pessoa que escreve à noite pode apagar a luz, mas as palavras que ela escreveu vão permanecer. O mesmo acontece com o destino que traçamos para nós neste mundo.”

Shãkyamuni

Agradecimentos

Primeiramente, à Deus, a quem recorri em tantas ocasiões e por inúmeras razões, agradeço pelas inspirações, pela força, pela disposição, pela perseverança, pela saúde e, principalmente, por ter me iluminado quando me achava perdido entre dúvidas e incertezas. Obrigado pelos desafios e pela coragem de enfrentá-los.

À meu orientador e amigo Ricardo Dahab, pela oportunidade, pelo incentivo, por sua valiosa orientação e, sobretudo pelo seu profissionalismo e dedicação durante o desenvolvimento desta dissertação. Agradeço mais uma vez a Deus pelo privilégio de tê-lo como meu orientador.

À minha noiva, Sandra, e à sua família pelo apoio e acolhimento. Sandra, a sua compreensão, ajuda e carinho foram e sempre serão fundamentais para mim. Agradeço e dedico este trabalho como uma pequena prova do meu amor por você.

À minha família pelo apoio em todas as etapas da minha vida; pelo amor, dedicação, carinho e, sobretudo, pelo apoio e incentivo que sempre me ofereceram, ajudando e respeitando minhas decisões de vida. Dedico também este trabalho aos meus pais Annayr e Euladio que fizeram e fazem muito por mim, sem eles este trabalho não existiria.

Aos amigos e colegas da SAP pela compreensão, companheirismo, sugestões de melhoria, pelas críticas construtivas e colaboração incondicional sempre que precisei.

Ao Instituto de Computação da Unicamp, seus professores e funcionários pela atenção, suporte técnico, amizade, contribuição intelectual e principalmente obrigado pela grande oportunidade.

À banca examinadora pelas sugestões.

E também a você que está lendo este trabalho!

Resumo

A indústria desenvolvedora de software vem promovendo a Arquitetura Orientada a Serviços (SOA) como sendo um novo marco no processo evolutivo de desenvolvimento de software. Segundo o Gartner Group, estima-se que a arquitetura SOA estará presente em grande parte dos novos desenvolvimentos de software ate o ano de 2008. Portanto, nesse processo evolutivo o estilo de arquitetar, desenvolver e implementar novos processos de negócios e aplicações serão fortemente influenciados pela arquitetura SOA.

Arquiteturas tradicionais se tornaram mais complexas, caras, difíceis de gerenciar e desafiadoras nos aspectos de integração, interoperabilidade e segurança. Como alternativa, surge a arquitetura SOA que promete reduzir custos integrando plataformas heterogêneas e reutilizando linhas de códigos já existentes, proporcionando assim agilidade na melhoria e ou geração de novos negócios.

A arquitetura SOA também tem seus desafios, talvez o maior deles seja disponibilizar mecanismos de seguranças adequados para cada tipo de negócio proporcionado por ela. Por exemplo, os processos de negócios baseados na arquitetura SOA tendem a ser diferentes dos processos de negócios baseados em arquitetura tradicionais, em conseqüência disso novos requisitos de segurança surgem e as vezes demandam mecanismos de segurança alternativos aos já existentes.

A arquitetura SOA esta de certa forma refém da capacidade de prover mecanismos de segurança adequados e principalmente interoperabilidade entre esses mecanismos. Esses são pelo menos os desafios mais relevantes que a arquitetura SOA tem pela frente antes de se tornar de fato uma arquitetura completa. Alias, o sucesso da Arquitetura Orientada a Serviço (principalmente serviços baseados em Web Services) esta totalmente dependente do sucesso na utilização dos mecanismos de segurança disponíveis.

Essa dissertação tem o propósito de discutir e avaliar as características básicas da Arquitetura Orientada a Serviços e os novos requisitos de seguranças gerados pelos processos de

negócios. A capacidade de interoperabilidade entre os mecanismos de segurança atuais e os novos mecanismos desenvolvidos para arquitetura SOA também fazem parte dessa dissertação.

Abstract

The software industry is promoting Service Oriented Architecture (SOA) as a new starting point of the evolutionary process of software development. Based on estimations from the Gartner Group, by 2008, most of the software development initiatives will be SOA-based. Hence, in this evolutionary process, SOA will be a strong influence in the design, development and implementation of new business processes and applications.

Traditional architectures became more complex, expensive, difficult to manage and challenging, with respect to the aspects of integration, interoperability and security. As an alternative, SOA architecture offers easy and cost-effective integration of heterogeneous platforms and the reuse of existing source code, with direct consequences in the dynamism of business.

SOA architecture also has its challenges: perhaps making available an adequate security mechanism for each type of business could be considered one of them. For instance, business processes based on SOA architecture usually are different from traditional business processes; therefore, new security requirements need to be addressed by security mechanisms, usually different from traditional ones.

SOA architecture is heavily tied with security mechanisms; providing interoperability between systems implies in finding a good coupling of different security mechanism, without losing the advantages provided by the architecture. Thus, these challenges have to be met before SOA becomes a complete architecture. In fact, the success of Services Oriented Architecture (mainly Web Services) is totally dependent on the success of its security mechanisms.

The objective of this dissertation is to discuss and evaluate the basic characteristics of the SOA architecture and the security requirements of business processes. The interoperability between traditional security mechanisms and the new security mechanisms of the SOA architecture are also covered by this dissertation.

Sumário

Lista de Figuras

Lista de Listas

Lista de Tabelas

Capítulo 1

1 Introdução.

O processo evolutivo de software, hardware e consequentemente das diversas aplicações embasadas nessas tecnologias, proporciona ao mercado diferentes maneiras e opções de processar informações e arquitetar novos modelos de negócios. De maneira geral o mercado é o maior beneficiado nesse processo evolutivo, e é também o responsável por impulsionar a indústria a desenvolver novas tecnologias que visam proporcionar vantagens que agregam valores aos processos de negócios como: agilidade de implementação, flexibilidade, segurança, padronização e interoperabilidade, redução de custos dos processos de negócios etc.

O surgimento da Arquitetura Orientada a Serviços é conseqüência deste processo evolutivo da tecnologia de software e hardware. A capacidade e o poder de processamento das informações juntamente com a velocidade em que as informações são trocadas, o alto nível de padronização dos protocolos, a abrangente infra-estrutura disponível considerando principalmente a Internet, proporcionou o momento adequado para difundir os conceitos da arquitetura SOA e atrair investimento em novos desenvolvimentos.

A arquitetura SOA promete inovar a forma de prover e consumir informações através de serviços, principalmente serviços disponibilizados na Internet. O aumento da popularidade nos últimos anos deve-se principalmente aos inúmeros benefícios que ela promete proporcionar aos processos de negócios, por exemplo, reutilização dos serviços, linhas de códigos e infra-estrutura já existentes. Essas são algumas funcionalidades que o mercado almejava em obter para viabilizar a redução dos custos com novos desenvolvimentos. Observando as necessidades do mercado e o momento adequado, a indústria de software e hardware uniu suas forças para definir normas, protocolos e especificações para tornar realidade a Arquitetura Orientada a Serviços.

Basicamente, a Arquitetura Orientada a Serviços consiste de serviços servidores ou consumidores que disponibilizam interfaces muito bem definidas que podem ser ativadas e

consumidas por outros serviços. Esses cenários com serviços servidores e consumidores não requerem necessariamente a infra-estrutura Web Services como muitos pensam, embora exista uma grande tendência na utilização dessa infra-estrutura como plataforma da arquitetura SOA. Em fato, é um erro comum achar que Arquitetura Orientada é um punhado de Web Services disponibilizados na Internet, ao contrário do que muitos pensam, a arquitetura SOA é um estilo de arquitetar e que vem ganhando espaço através de Web Services.

Considerando todos os benefícios e quebra de paradigma que a Arquitetura Orientada a Serviços promete proporcionar para o mercado, juntamente com os prognósticos de crescimento até 2008 apontados pelo Instituto Gartner [KOD05], faz surgir a pergunta se a arquitetura SOA dispõe de todos os mecanismos de segurança, infra-estrutura e principalmente mecanismos de interoperabilidade necessários para proporcionar um ambiente seguro baseado nos novos requisitos de segurança que surgem nos novos processos de negócios baseados na arquitetura.

A padronização recente das normas, protocolos e especificações, bem como a crescente demanda estimada no curto prazo da arquitetura SOA, oferecem embasamento para preocupação com o potencial da indústria e provedores de serviços a disponibilizarem mecanismos de segurança adequados para garantir a segurança e interoperabilidade nos processos de negócios embasado na arquitetura SOA. A preocupação com segurança e interoperabilidade é válida uma vez que parte do processo de implementação da arquitetura é a relação de confiança entre provedor e consumidor de serviços e a disponibilidade das funcionalidades oferecidas pelas recentes especificações.

Um artigo escrito por Ray Wagner diretor de pesquisa do departamento de segurança estratégica do Gatner, afirma que "The successful deployment of standards-based security technologies will be a key determinant in the widespread adoption of Web services" [WSI05]. Em resumo, ele comenta que o fator determinante da adoção em massa de Web Services depende do sucesso da utilização de tecnologias de segurança baseados em padrões. Ou seja, existe um consenso que a segurança é um fator extremamente importante e estratégico para impulsionar ou alavancar uma utilização em massa de Web Services.

1.1 Motivação.

Diante da grande popularidade da Arquitetura Orientada a Serviços na mídia, dos prognósticos que apontam a sua adoção em massa até o ano de 2008 e principalmente dos aspectos de segurança diferenciados como, por exemplo, a complexidade de implementação dos novos mecanismos de segurança, da falta de documentação e casos de sucessos, informação e estudo sobre assuntos relacionados à segurança, infra-estrutura disponível, os novos protocolos e padrões, interoperabilidade etc., motivaram a elaboração dessa dissertação na qual é feito um estudo sobre os desafios de segurança existentes na adoção da Arquitetura Orientada a Serviços.

É fato que a tecnologia vem avançando rapidamente nos últimos anos, e que cada vez mais o mercado se aproveita deste avanço para aprimorar seus processos de negócios, tornando-se mais competitivos e lucrativos. Atualmente, empresas utilizam a infra-estrutura que possuem para oferecer diferentes tipos de serviços sejam eles via web, conexão dedicada, VPN, serviços locais cliente/servidor etc. Em resumo, toda essa infra-estrutura atual que vemos são conexões altamente acopladas em sua maioria, onde as regras do jogo entre dois pontos são pré-estabelecidos e a inter-conectividade é geralmente feita entre duas interfaces apenas (Cliente/Servidor) o que torna razoavelmente fácil gerenciar pequenas infra-estruturas embora limite a interoperabilidade entre processos.

Com todo esse marketing, prognósticos de crescimento e euforia crescente do mercado observado em torno da Arquitetura Orientada a Serviços, observamos o preocupante fato que segurança da informação está sendo deixada também para segundo plano como vem acontecendo na maioria dos desenvolvimentos e implementações atuais, por exemplo, Enterprise Application Integrator - EAI. Isso significa que as empresas de forma errônea estão implementando ou amadurecendo a idéia de implementar a arquitetura SOA sem considerar, em paralelo, os aspectos de segurança relevantes para o sucesso do negócio. Isso é preocupante e pode colocar em risco o processo de negócio da empresa e o próprio sucesso de crescimento arquitetura SOA esperada pela indústria.

1.2 Objetivos deste trabalho

Essa dissertação tem o propósito de avaliar os mecanismos de segurança atualmente disponíveis para Arquitetura Orientada a Serviços. O objetivo é identificar os requisitos de segurança demandados pelos processos de negócios baseados na Arquitetura Orientada a Serviços e discutir os desafios de segurança existentes. Por exemplo, garantia de interoperabilidade entre mecanismos de segurança utilizando certificados digitais no processo de assinatura digital e criptografia, proporcionar infra-estrutura para oferecer interoperabilidade entre os mecanismos de segurança, considerar o risco com a utilização do modelo centralizado de gerenciar a relação de confiança etc.

Também é objetivo dessa dissertação mostrar as diferenças existentes nos aspectos de segurança entre as arquiteturas atuais e a arquitetura SOA. Por exemplo, um modelo de negócio baseado em Web Services através da aplicação Enterprise Application Integrator (EAI) será ilustrada nessa dissertação para exemplificar os requisitos de segurança envolvidos nos processos de negócios proporcionados por ela. Com a ilustração, poderemos então observar e avaliar as diferenças nos aspectos de segurança existentes entre negócios baseados em aplicações EAI e negócios baseados na arquitetura SOA.

1.3 Abordagem seguida neste trabalho

O trabalho se dividiu em duas partes principais: a descrição e objetivo da Arquitetura Orientada a Serviços com suas características e particularidades, e os mecanismos de segurança disponíveis associados aos desafios esperados na implementação da Arquitetura Orientada a Serviço.

A primeira parte do trabalho buscou tanto na indústria, através de “white papers” e sites na Internet como W3C, IETF entre outros, quanto na literatura acadêmica, através de livros, artigos, dissertações e teses, uma fundamentação para a definição da Arquitetura Orientada a Serviços e suas particularidades.

A segunda parte se baseou fortemente nos conhecimentos pessoais resultantes de vários anos de experiência na área de consultoria em segurança da informação e também contou com uma exaustiva procura (Internet, livros, artigos etc.) por informações relativas aos mecanismos de segurança disponíveis para arquitetura SOA. É importante mencionar que na procura por informações referente aos aspectos de segurança SOA, observou-se que não existe muita informação tratando assuntos de segurança na utilização da Arquitetura Orientada a Serviços ou bem como relatos de casos de sucesso de implementação.

1.4 Organização deste trabalho.

Esta dissertação possui cinco capítulos. No capítulo 2 será mostrado o que é uma arquitetura SOA detalhando suas características, vantagens e desvantagens, bem como alguns aspectos de segurança relevantes somente a essa arquitetura. Após, no capítulo 3, serão mostrados os mecanismos de segurança existentes atualmente e como eles poderão estar sendo utilizados juntamente com a arquitetura SOA. O Capitulo 3 faz uma análise dos requisitos de segurança gerados pelos processos de negócios baseados na Arquitetura SOA, e os mecanismos de segurança desenvolvidos especialmente para atender algumas demandas específicas.Uma abordagem sobre a capacidade de interoperabilidade entre os mecanismos de segurança também é tratada no capitulo 3. Finalmente no capítulo 4 serão feitas as considerações finais e apresentadas algumas possibilidades de trabalhos futuros.

Neste trabalho vamos observar que ora se faz referência a Arquitetura Orientada a Serviços e ora se faz referência a arquitetura SOA, os dois termos estão corretos e significam a mesma coisa e serão utilizados nessa dissertação. A Arquitetura Orientada a Serviços é a versão em português do termo traduzido do inglês Service Oriented Architecture. No Brasil existe certa padronização na utilização destes dois termos na versão em português como estaremos utilizando nessa dissertação.

Capítulo 2

2 Arquitetura Orientada a Serviços.

Como um dos assuntos com maior popularidade atualmente no mercado de software, a Arquitetura Orientada a Serviço vem sendo considerada como um marco na evolução de software. Tal popularidade foi alcançada em função das possibilidades de arquitetar e desenvolver novas aplicações e processos de negócios, reaproveitando principalmente códigos já existentes e plataformas heterogeneas, bem como vantagens estratégicas proporcionadas pela arquitetura. Unindo essa popularidade com os beneficios e vantagens proporcionadas por ela, SOA se torna objeto de desejo de inúmeras empresas dispostas a sair na frente com as tentadoras vantagens oferecidas.

A arquitetura SOA é também alvo de inúmeras previsões efetuadas por entidades conceituadas de mercado como o respeitado Gartner Group. Recentemente, uma reportagem do Gartner Group [Gartner Group] faz uma análise dos 5 tópicos mais discutidos no ano de 2005, que aponta que em 2008 80% dos novos projetos de desenvolvimentos de Software serão baseados na Arquitetura Orientada a Serviços [KOD05].

2.1 O que é Arquitetura Orientada a Serviços?

A Arquitetura Orientada a Serviços, é composta por um conjunto de conceitos e regras que proporciona a base para arquitetar, desenvolver sistemas e aplicações orientadas a serviços [CEP03] visando obter o máximo de desacoplamento (loose coupling) entre serviços. Muitos desses conceitos e regras existentes na arquitetura SOA, foram baseados em modelos já existentes como, por exemplo, processamento distribuídos, orientação a objetos (DCOM, CORBA etc.) entre outros [CAP05].

A arquitetura SOA pode consistir de vários serviços que atuam como serviços consumidores e/ou serviços provedores. De maneira geral esses serviços ficam disponíveis para que outros serviços os acessem, caracterizando assim como serviços consumidores e serviços provedores dependendo de quem consome ou provê o serviço. A arquitetura SOA também permite composições de serviços, ou seja, serviços consumidores que consomem serviços provedores que consequentemente se tornam serviços consumidores de outros serviços provedores [SER05]. Composição de serviços teoricamente será muito comum na arquitetura SOA, em consequência disto mecanismos de segurança adequados deverão estar presente para garantir segurança e sigilo deste novo modelo de negócio.

Em agumas reportagens equivocadas, a Arquitetura Orientada a Serviços (SOA) é referenciada por alguns especialistas de tecnologia como sendo um conjunto de Web Services disponíveis na Internet. Embora essa afirmação não esteja totalmente errada, devemos frisar que a arquitetura SOA independe de ambiente Web Services para existir, ou seja, SOA é um modelo conceitual de arquitetura orientada a serviços, e não apenas um produto, aplicação ou módulo de sistema a ser implementada. O importante é entender que a arquitetura SOA é baseada em protocolos, conceitos, regras, padrões e acordos contratuais para troca de informação entre serviços, e que Web Services esta se tornando de fato a plataforma preferida para se implementar a arquitetura SOA.

2.1.1 Arquitetura SOA comparada com a rede mundial Internet.

A arquitetura SOA é constituída basicamente pela troca de mensagens SOAP [LAT05] entre serviços através da utilização dos protocolos open Standards XML, WSDL [WEB05], SOAP [LAT05], UDDI [UDD05] etc. Para um melhor entendimento da arquitetura SOA bem como algumas de suas características, vamos fazer uma comparação entre a arquitetura SOA e a rede mundial Internet para ilustrarmos as semelhanças entre as principais características das duas arquiteturas.

Na Internet temos os Web Servers (provedores de páginas html e serviços) e os Browsers/usuários (consumidores de páginas html e serviços). Os Web Servers (provedores de páginas html serviços) e Web Browsers/usuários (consumidores de páginas html e serviços) não estão necessariamente conectados entre si todo momento. Os serviços providos pelos Web Servers também não são previamente conhecidos ou dependentes de plataforma hardware ou software para estabelecer uma conexão. Sendo assim, podemos notar que na rede mundial de Internet encontramos parte da característica básica e fundamental da arquitetura SOA, o acoplamento fraco de serviços (Loose Coupling) que na prática significa que qualquer modificação nas páginas e serviços oferecidos não afetará o acesso.

A troca de informação na rede mundial Internet é feita basicamente através dos protocolos HTTP e páginas HTML. A troca de informação na arquitetura SOA é feita através das mensagens XML/SOAP sobre o protocolo HTTP. Para garantir a segurança dos dados trafegados entre os serviços na rede mundial Internet, são utilizados os protocolos SSL/TLS que garantem segurança do transporte, e dependendo do requisito de segurança exigido na troca dessas informações, podem-se utilizar certificados digitais para assinar digitalmente ou cifrar os dados trafegados. Já na arquitetura SOA os mecanismos de segurança ganharam outros aliados, por exemplo, XML digital signature e XML encryption, que garantem a segurança da mensagem.

2.1.2 Enterprise Application Integrator - EAI.

Em um modelo de negócio mais complexo via Web Services, estaremos utilizando os protocolos HTTP, XML, SOAP, WSDL, etc. para troca de mensagens via Enterprise Application Integrator (EAI). A utilização de EAI para troca de mensagens via Web Services é feita de maneira totalmente acoplada, ou seja, para que haja comunicação entre os EAIs os mesmos devem conhecer antecipadamente a interface que irão acessar e qualquer modificação na interface pode causar problemas de comunicação. Neste caso se faz necessário na maioria das vezes a troca de arquivos WSDL que descreve a estrutura das interfaces de cada serviço, comunicação de possíveis modificações nas interfaces, bem como informações sobre mecanismos de segurança a serem utilizados, por exemplo, SSL/ TLS, Certificado Digital, Assinatura Digital etc.

Comentário.

A utilização *de EAI é a forma atual que a maioria das empresas adotou para prover e consumir serviços entre parceiros de negócios ou mesmo prover serviços dentro de suas próprias infra-estruturas interligrando sistemas heterogêneos. A questão de segurança na maioria dos casos utilizando EAI é tratada também separadamente pelas empresas, primeiramente as empresas disponibilizam seus Web Services, e em segundo plano tentam acoplar os mecanismos de segurança conforme exigência do negócio. Na arquitetura SOA já percebemos que a segurança também esta ficando para segundo plano, mas o fator complexidade da arquitetura pode colocar em risco a implementação como um todo se os aspectos de segurança não forem abordados no início de sua implementação.*

Embora exista uma semelhança na utilização dos protocolos e mecanismo de segurança, a Arquitetura SOA e Enterprise Application Integrator (EAI) são diferentes, no entanto, SOA pode ser visualizado com uma evolução do EAI [ANA05]. O mercado atualmente esta cada vez mais utilizando EAIs para integrar sistemas através de Web Services via Internet. A grande tendência de agora em diante será a utilização dos conceitos da arquitetura SOA para facilitar e inovar a maneira de prover esses Web Services mais eficientemente.

2.1.3 Ilustração de cenário de negócios utilizando EAI.

Neste cenário de negócios ilustrado logo a baixo temos duas empresas fictícias, empresa A que utiliza Enterprise Resource Planning – ERP da PeopleSoft e empresa B que utiliza Enterprise Resource Planning – ERP R/3 da SAP. As duas empresas estão geograficamente separadas por uma grande distância, porém trocam informações de negócios via Internet para reduzir os custos com linha privada etc. Essas informações trocadas podem ser das mais diversas. Assim, por exemplo, se as duas empresas pertencem ao mesmo grupo de acionistas as informações trocados poderiam ser referente a informações financeiras de cada uma delas. As

informações deste cenário ilustrado também poderiam ser entre clientes e fornecedores, empresas e bancos etc.

Observamos que empresa A tem um ERP diferente da empresa B, portanto estabelecer uma interface de comunicação diretamente entre ERPs seria quase impossível se considerarmos as diferenças na maneira que os dados são interpretados e tratados por cada sistema ERP. Justamente para resolver esse tipo de questão de interoperabilidade e integração entre sistemas que empresas optam por utilizar EAI para agir como intermediário da comunicação e troca de dados entre diferentes sistemas.

O EAI conforme ilustrado atua meramente como integrador entre sistemas, ou seja, enquanto a arquitetura SOA é baseada em conceitos, regras, contratos etc., o EAI se restringe basicamente a prover serviços totalmente acoplados de integração entre sistemas. Os protocolos de comunicação HTTP, XML, SOAP, WSDL, etc. utilizados pelo EAI são os mesmos utilizados na arquitetura SOA, porém os conceitos, regras contratos fazem com que a arquitetura SOA seja um modelo totalmente inovador de prover serviços se comparado com o modelo de integração de sistemas utilizado pelos EAIs.

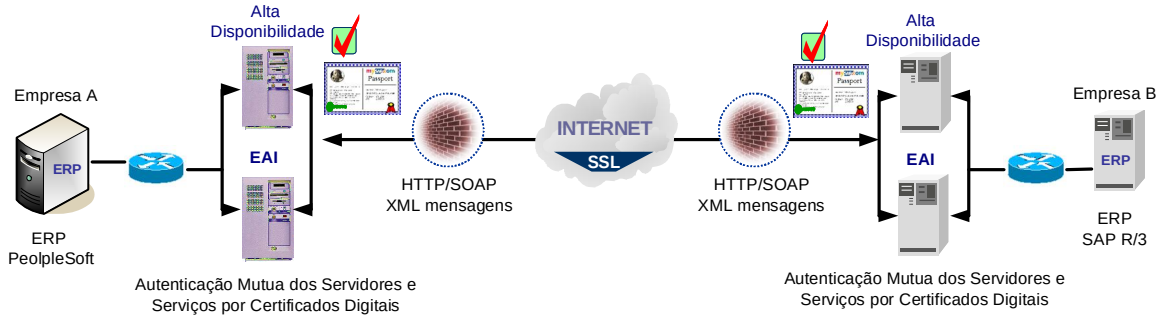


Figura 01 - Troca de dados via EAI.

Comentário.

Nesta ilustração, os protocolos e dispositivos de segurança (SSL/TLS, Certificado Digital e Assinatura Digital) garantem a segurança e o sigilo das informações na

comunicação ponto a ponto na camada de transporte. Os mecanismos de segurança são definidos previamente entre as partes envolvidas com o auxílio das pessoas envolvidas no processo de implementação da comunicação entre os EAIs. Os serviços são totalmente acoplados, isso significa um aumento no risco de conflitos entre serviços quando modificações dos mesmos forem necessários. As interfaces de serviços também devem ser previamente conhecidas para que seja possível a comunicação.

2.2 O Relacionamento entre Web Services e a Arquitetura SOA.

Como já foi mencionado anteriormente, Web Services não é requisito para se obter, implementar ou estabelecer uma arquitetura SOA. O Gartner Group [GAR05] fez um relatório definindo justamente as diferenças entre essas duas tecnologias para esclarecer essa confusão do relacionamento entre Web Services e arquitetura SOA. Em Abril de 2003 Yefim V. Natis do Gartner descreveu o seguinte: "Web services são baseados em especificações tecnológicas, enquanto a arquitetura SOA é baseado em princípios de desenvolvimento de software. De forma notável, Web Services Description Language (WSDL) utilizado por Web Services é o padrão que define interfaces para arquitetura SOA, este é o momento que Web Services e a arquitetura SOA fundamentalmente se conectam." [NAT03]

"Os *Web services* são componentes que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato *XML*. O W3C e o OASIS são as instituições responsáveis pela padronização dos Web services. Empresas como IBM e Microsoft, duas das maiores do setor de tecnologia, apóiam o desenvolvimento deste padrão" - [WEBD06].

Em resumo, a arquitetura SOA é um estilo de projetar sistemas, enquanto Web Services são serviços implementados através da utilização de padrões. Web Services é uma das maneiras

que podemos implementar a arquitetura SOA [NAT03]. Os benefícios de uma arquitetura SOA através de Web Services são alcançados pelo acesso a serviços independentes de plataformas, interoperabilidade em função do grande suporte proporcionado pela industria associada aos inúmeros padrões voltados para Web Services [KOD05].

2.3 Objetivo da Arquitetura Orientada a Serviços.

Uma implementação SOA tem o objetivo básico de proporcionar a facilidade de reutilização, integração, flexibilização de serviços e códigos já oferecidos dentro de uma infra-estrutura corporativa, proporcionando assim a redução de geração de novos códigos e serviços redundantes. Para alcançar esses objetivos propostos pela arquitetura SOA, vários requisitos inerentes a arquitetura devem ser observados. Por exemplo, os serviços provedores e consumidores não devem ter dependências tecnológicas para se comunicarem; ou seja, eles devem ser independentes de plataforma que estejam utilizando para que a comunicação seja sempre possível.

Outra característica muito importante da arquitetura SOA é o acoplamento fraco (Loose Coupling), isto significa que os serviços provedor e consumidor devem reduzir ao máximo as possíveis interdependências das interfaces, característica essa que não ocorre na utilização de EAIs observados na nossa ilustração de utilização EAI (Figura 01). A característica de (Loose Coupling) refere-se à maneira de desenvolver serviços de forma a proporcionar a redução significativa das interdependências entre os mesmos. A ideia é possibilitar alterações e inclusões de novas funcionalidades e até exclusão do serviço sem causar conflitos a outros serviços.

Em resumo, o objetivo fundamental da arquitetura SOA e proporcionar uma arquitetura que flexibilize o reuso e integração de serviços e códigos já existentes, evitar as práticas tradicionais de desenvolver novos códigos para prover funcionalidades as vezes já existentes dentro de um ambiente corporativo. Argumentos como de reutilização de códigos e serviços já existentes fez com que a arquitetura SOA tornasse popular e atraente para as empresas.

Benefícos esperados como ROI Retorno no Investimento (**R**eturn **o**f **I**nvestiment) [GRA05], agilidade no desenvolvimento, ganho de competitividade, flexibilidade, entre outros ajudou a alavancar no mercado um grande entusiasmo e expectativa da arquitetura SOA. No entanto, vale frisar que o sucesso da arquitetura SOA esta totalmente dependente do sucesso na utilização dos mecanismos de segurança adequados para proporcionar e manter essa tão desejada e almejada arquitetura promissora.

2.4 Entidades da Arquitetura Orientada a Serviços.

Arquitetura Orientada a Serviços (SOA) é praticamente composto por várias funcionalidades conceituais e fundamentais, e quando implementadas juntas proporcionam embasamento para o paradigma do **find**, **bind e execute**.

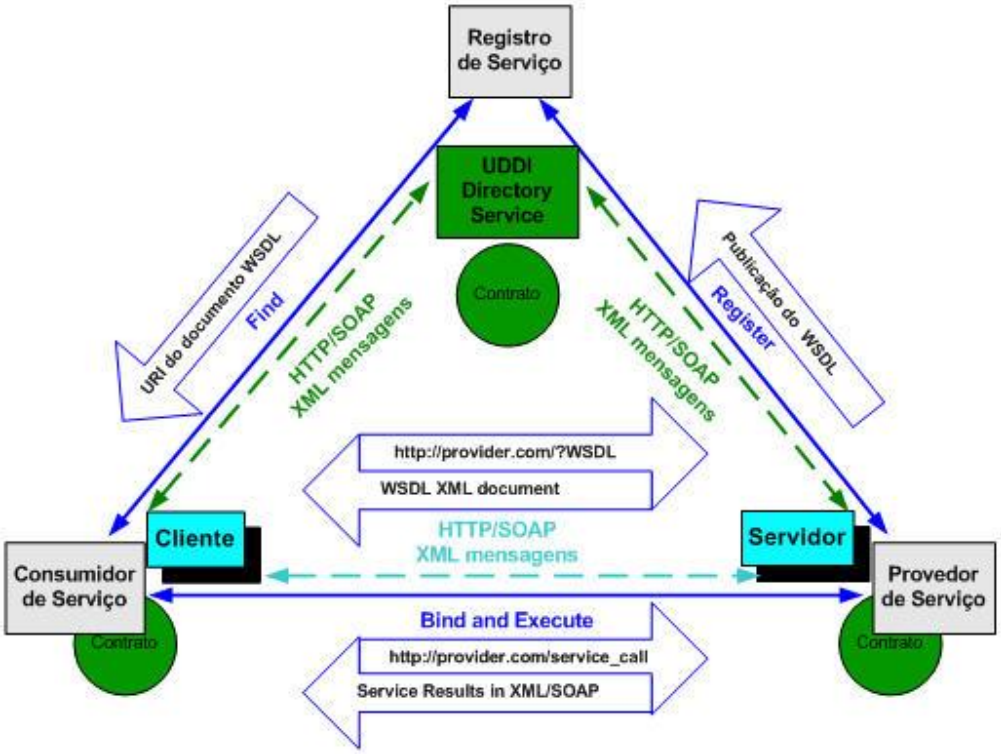


Figura 02 - O paradigma de "find-bind-execute".

A Figura 02 acima ilustra a arquitetura SOA com todas as suas entidades básicas e conceituais, porém em muitos casos essas entidades conceituais nem sempre estão presentes ao mesmo tempo na arquitetura. Por exemplo, a entidade Registry onde são armazenados a localização dos serviços e dos respectivos contratos, em sua maioria pode não ser utilizada na arquitetura SOA como ilustrado na figura.

Cada entidade dentro da arquitetura SOA tem sua funcionalidade especifica, e as interações entre essas entidades ocorrem de forma a proporcionar o máximo de desacoplamento. Para um melhor entendimento dessas funcionalidades, será feita uma breve introdução a seguir da característica de acoplamento fraco (Loose Coupling) e de cada entidade SOA presente na Figura 02.

2.4.4 Acoplamento fraco ou Loose Coupling.

Acoplamento fraco (Loose coupling), é uma característica de sistema que referência uma crescente tendência na maneira de projetar interfaces provedoras de serviços. Acoplamento fraco tem a finalidade de reduzir a interdependência e possíveis riscos de conflitos entre as entidades. O propósito de projetar interfaces com o máximo de desacoplamento possível (Loose coupling), é proporcionar flexibilidade de incluir, adicionar e modificar interfaces com o mínimo ou nenhum impacto de interoperabilidade entre as interfaces já existentes.

Uma tabela muito interessante retirada de um artigo escrito por Doug Kaye [KAY05], faz uma excelente comparação entre sistemas acoplados e sistemas desacoplados. Esta tabela, segundo descreve o autor, sofreu varias modificações através de comentários que recebeu dos leitores que tinham pontos de vista diferentes sobre o assunto. A tabela comparativa entre os modelos acoplados (Tightly Coupled) e modelos desacoplados (Loosely Coupled) ajuda desenvolvedores que buscam desenvolver ou arquitetar sistemas com máximo de desacoplamento possível entre interfaces, requisito este fundamental para uma arquitetura baseada em SOA.

Comentário

Sistemas, aplicações e novos desenvolvimentos baseados na arquitetura SOA estão proporcionando desafios de segurança uma vez que todo o processo de comunicação entre serviços são dinâmicos conforme características da arquitetura. Os desafios de segurança na arquitetura SOA podem ser de pequena, média e grande complexidade dependendo do modelo de negócio implementado, e ou requisito mínimo de segurança exigido. Por exemplo, se os requisito por segurança exigirem certificação digital nos processos de negócios, os desafios para estabelecimento de relação de confiança para validação dos certificados digitais podem ser considerados como desafio de média a grande complexidade.

	Tightly Coupled	Loosely Coupled
Interaction Style	RPC	Document
Granularity	Object	Message
Synchronization	Synchronous	Asynchronous
Technology Mix	Homogeneous	Heterogeneous
Data Types	Dependent	Independent
Syntactic Definition	By Convention	Published Schema
Bindings	Fixed and Early	Delayed
Semantic Adaptation	By Re-coding	Via Transformation
Software Objective	Re-use, Efficiency	Broad Applicability
Consequences	Anticipated	Unintended

Tabela 01 – Comparação Arquitetura Fortemente acoplada e Fracamente acoplada, [KAY05]

Esta tabela acima nos ajuda a efetuar uma comparação das semelhanças e diferenças entre a arquitetura SOA (desacoplado) e o nosso exemplo de Web Services utilizando EAI (acoplado) ilustrado na Figura 01. Por exemplo, a definição sintática para comunicação entre interfaces na

arquitetura SOA é feita dinamicamente través de publicação do endereço de localização do esquema WSDL em um diretório de serviços. No caso de Web Services utilizando EAI o esquema tem que ser previamente conhecido e configurado manualmente para que seja possível a comunicação. As semelhanças também existem entre SOA e EAI, por exemplo, os dois proporcionam conectividade em ambientes heterogêneos embora a tabela mostre o contrário.

2.4.5 Consumidor de Serviço ou Service Consumer.

Um consumidor de serviço pode ser proporcionado por qualquer módulo de software que requer serviços [STE05], seja ele aplicação, serviço etc. O importante é lembrar que não é necessário ser um módulo ou componente web para ser considerado parte da arquitetura SOA. Consumidor de serviço dentro dos conceitos SOA é uma entidade que efetua a busca no repositório de registro de serviços (registry) para localizar determinado serviço. Uma vez encontrado, a entidade consumidor de serviço dispara uma requisição de abertura de conexão contra a interface de serviço procurado conforme formato e regras de contratos exigidos pelo serviço provedor.

A falta da entidade de registro (registry) na arquitetura como mencionamos anteriormente não causará impactos. Porém de alguma forma o consumidor de serviço terá que descobrir como chegar ao serviço desejado. Pode-se utilizar, por exemplo, um repositório LDAP, ou simplesmente apontar para um endereço fixo o qual pode ser até mesmo do provedor de serviços que se deseja acessar.

2.4.6 Provedor de Serviço ou Service Provider.

Um provedor de serviço pode ser disponibilizado por qualquer módulo de software ou sistema que executa requisições de serviços provenientes de consumidores de serviços. A entidade provedor de serviço é um serviço endereçável que aceita e executa requisições enviadas

por consumidores de serviços desde que as requisições obedeçam regras de contrato e descrições do serviço contidas no arquivo WSDL. Os detalhes dos serviços providos pela entidade provedor de serviço provedor, são públicados em um repositório de registro (registry) juntamente com o a localização do contrato que contém regras e requisitos de segurança para acessos aos serviços providos [STE05].

2.4.7 Registro de Serviço ou Registry.

A entidade registro de serviço (registry) funciona como um diretório de rede ou como uma espécie de DNS (Domain Name Service) que utilizamos na rede mundial Internet para localizar páginas html, Web services etc. Porém, ao invés de resolver nomes em endereços IPs e prover a localização das páginas html, Web services etc, o registro de serviço contém a localização dos serviços disponíveis (basicamente aponta a localização do WSDL do serviço procurado), bem como prove a localização dos contratos e requisitos de segurança.

Uma das características básicas do registro de serviço é determinar o tipo de segurança e protocolo suportado pelos serviços requisitados, e prover a localização dos WSDLs e políticas de acessos (contratos) dos mesmos para consumidores de serviços que os requisitam. O padrão UDDIv3 define o protocolo de comunicação entre consumidores e servidores de serviços com o registro de serviço [STE05]. A última versão do protocolo UDDI também suporta assinatura digital.

2.4.8 Contrato de Serviço ou Service Contract.

Um contrato é uma especificação do provedor de serviço que fornece ao consumidor de serviço detalhes para a estrutura de requisição do serviço e detalhes da estrutura de resposta que o provedor de serviço proporcionará. O contrato é responsável por passar detalhes como qualidade de serviço (QoS), por exemplo, pode determinar o tempo limite que um consumidor de serviço

pode consumir um serviço provido por um provedor de serviço, isso proporciona a redução do tempo de acoplamento entre serviços.

Os contratos também podem trazer condições especiais estipuladas pelo provedor de serviço para execução de determinada funcionalidades. Políticas de segurança também são outro exemplo do que os contratos podem proporcionar, por exemplo, autenticação, sigilo etc. Segue abaixo um resumo dos possíveis itens que um contrato de serviço pode oferecer [STE05]:

Lista 01 - Itens de contrato de serviço

- Lista de prováveis condições excepcionais.
- Requerimentos de segurança, incluindo criptografia e assinatura digital.
- Informação de qualidade de serviço (QoS) que descreve métricas.
- Etc.

2.4.9 Proxy de Serviço or Service Proxy.

O proxy de serviço pode ser comparado aos servidores de proxy de acesso à rede mundial Internet. Assim como um servidor de proxy para Internet proporciona uma significativa melhora na performance e segurança dos usuários de internet, o proxy de serviço para arquitetura SOA também proporciona maior performance para os consumidores de serviços e também aumenta a segurança. A principal característica que difere o proxy desenvolvido para arquitetura SOA dos demais, é que o proxy para arquitetura SOA foi desenvolvido para atender aplicações XML. Por exemplo, um processo de negócio baseado na arquitetura SOA que não requer dados atualizados em tempo real, pode simplesmente se beneficiar do armazenamento local de um serviço no proxy de serviço, proporcionando assim um ganho no tempo de resposta do serviço solicitado [STE05].

Comentário.

O proxy de serviço não é um requisito na arquitetura SOA como percebemos na Figura 02, embora sua presença aumente significativamente o ganho de performance e pode proporcionar um ambiente mais seguro. Outros componentes como roteadores

XML, firewalls XML etc. também estão surgindo com o propósito de suprir a demanda provocada pela arquitetura SOA e principalmente oferecer segurança para aplicações baseadas em XML. Esses novos dispositivos proporcionam, por exemplo, que mensagens SOAP sejam roteadas com base no seu conteúdo, e poderá também ser bloqueada caso o seu conteúdo seja identificado como conteúdo malicioso ou suspeito. Segue abaixo a Figura 03 ilustrando a utilização do proxy de serviço entre o registro, consumidor e provedor de serviços.

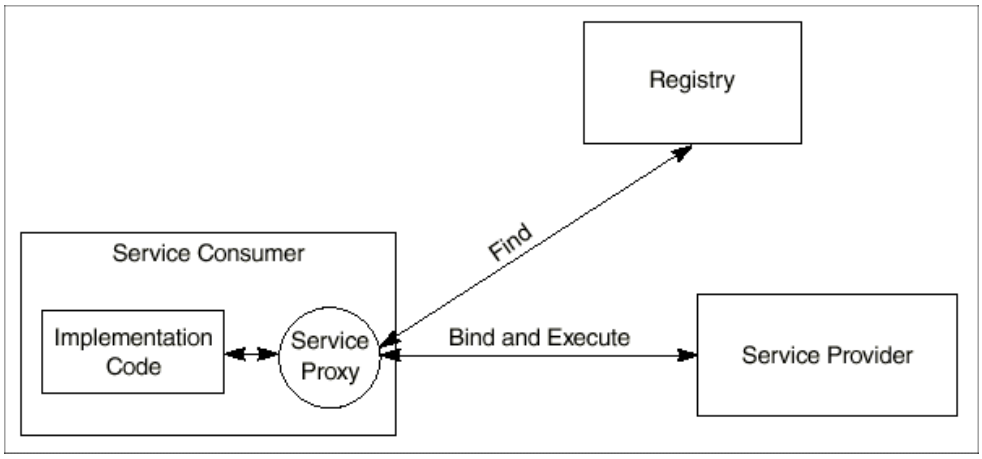


Figura 03 - A service proxy, [STE05]

2.5 Característica SOA.

Encontramos inúmeras características que podem descrever os atributos da arquitetura SOA. Certamente, não existe um número exato de características que uma arquitetura SOA deve conter, porém algumas características são fundamentais para identificar e caracterizar uma arquitetura SOA, por exemplo, plataforma independente para os serviços (**Platform-independent Interface**), localização dinâmica dos serviços (**Dynamic Discovery**), serviço com interface

endereçável (**N**etwork-addressable Interface), serviços desacoplados (**Loose Coupling**), serviço com interface auto-descritiva (**Self-describing Interfaces**) entre outros [STE05].

2.5.1 Independência de Plataforma.

Independência de plataforma aplica-se a provedor de serviço, consumidor de serviço e registro de serviço. Isso significa que não deve existir dependência de plataforma tecnológica (Hardware, Software) para as funcionalidades de localizar, prover ou consumir serviços em uma arquitetura SOA. Todos os novos desenvolvimentos baseados na arquitetura devem levar em consideração essa característica de independência de plataforma. Essa característica pode ser facilmente alcançada utilizando ferramentas de desenvolvimento que proporcionem a utilização de padrões abertos.

2.5.2 Localização dinâmica.

Localização dinâmica dos serviços aplica-se o registro de serviço (**U**niversal **D**escription, **D**efinition, and **I**ntegration - **UDDI**), que é o padrão disponibilizado registro de serviço. O registro de serviço deve proporcionar mecanismos de localização dos serviços oferecidos e requeridos pela entidade consumidora de serviços baseando-se em critérios. Por exemplo, um consumidor de serviço faz uma busca no servidor de registro de serviço para localizar provedores de serviços que forneçam notícias e informações locais de eventos. Baseando-se no tipo de notícia que se busca pode-se também efetuar uma filtragem por data, cidade desejada, estado e país, onde ocorreu a notícia ou acontecerá o evento etc.

Comentário.

Localização dinâmica de serviços ocorre através dos servidores de registros de serviços que atualmente não são encontrados em grande escala no mercado.

Esses serviços podem vir a serem oferecidos para o mercado em forma de serviços pagos inviabilizando o modelo de negócio que visa redução de custos. A localização dinâmica proporcionada pelo registro de serviço também pode sofrer com problemas de interoperabilidade entre os mecanismos de segurança. A interface consumidora e a interface servidora terá de alguma forma de confiar uma na outra caso a política de autenticação do serviço for, por exemplo, através de certificados digitais.

2.5.3 Interface endereçável.

Serviço com **interface endereçável** é a forma de localizar um serviço disponível na rede. Interface endereçável está necessariamente ligada a entidade de registro de serviço, de maneira que quando uma entidade consumidora de serviço executa uma busca no registro de serviço, o mesmo retorna o endereço Internet Protocol - IP [INT06] de rede para o consumidor de serviço localizar o serviço requisitado. Geralmente este endereço retornado aponta diretamente para o endereço de localização do **W**eb **S**ervices **D**escription **L**anguage - **WSDL** do serviço.

Comentário.

A interface endereçável proporcionou flexibilidade de comunicação entre os novos modelos de negócios baseados na arquitetura SOA. As políticas de segurança requisitadas por essas interfaces podem causar situações graves de interoperabilidades como já foi mencionado no comentário anterior. Embora a exista um protocolo denominado XKML (abordado no capítulo III) que visa tratar desse problema, ainda não esta claro como será abordada essa questão sem intervenção manual para não enfraquecer a segurança e sem perder a característica importante de interface endereçável.

2.5.4 Desacoplamento.

Como já foi mencionado no item 2.4.4, acoplamento fraco refere-se basicamente ao número de dependências existentes entre as entidades, serviços ou interfaces. Além disso, observamos que existe o termo “tight Coupled” ou totalmente acoplado, ou seja, serviços, entidades ou interfaces totalmente dependentes e acopladas. Na arquitetura SOA o objetivo é proporcionar interfaces desacopladas, os serviços consumidores e serviços provedores devem conter o mínimo possível de dependências entre si para possibilitar um melhor gerenciamento das interfaces caso houver a necessidade de alteração da mesma.

2.5.5 Interface auto-descritiva.

Serviço com **interface auto-descritiva** é proporcionado pelo padrão **W**eb **S**ervices **D**escription **L**anguage- **WSDL** utilizado para descrever os servidos providos. O WSDL funciona como uma gramática da plataforma utilizada. Atualmente em aplicações EAI é necessário efetuar manualmente a importação do WSDL da interface do serviço a ser contatado. Já no modelo SOA os serviços estarão aptos a prover interfaces auto-descritivas que proporcionaram a leitura do WSDL dinamicamente; ou seja, não haverá a necessidade de efetuar a troca manual do WSDL antecipadamente como é feito atualmente com o modelo Web Services via EAI.

Comentário.

É importante notar que na arquitetura SOA a interface é auto-descritiva e que na aplicação EAI é exigido uma troca prévia de arquivo de descrição denominado WSDL. Essa auto-descrição é peça fundamental para tornar processos de negócios flexíveis, no entanto, a política de acesso imposta pelos requisitos de segurança pode causar problemas de interoperabilidade para interfaces que exigem autenticação mútua, por exemplo. Contornar a situação com intervenção manual pode acarretar custos, enfraquecer a segurança, engessar o negócio etc.

2.6 Vantagens e Desvantagens da Arquitetura Orientada a Serviços.

Como qualquer outra tecnologia, a Arquitetura Orientada a Serviços também possuem suas vantagens e desvantagens. Diversos segmentos do mercado (Indústria, Comércio e Segmento de Serviços), estudam adotar a arquitetura SOA almejando vantagens como, aumento da competitividade, melhoria da eficiência das operações existentes, ganhar dinamismo nos processos para aproveitar as novas oportunidades de negócios, utilizar a infra-estrutura já existente, reutilizar os códigos existentes, reduzir custos etc.

Por se tratar de uma arquitetura tecnológica relativamente recente e com pouca abrangência no mercado atual, ninguém arrisca em apontar ou sugerir possíveis desvantagens que a arquitetura SOA pode proporcionar.

Comentário.

No entanto, eu acredito que uma das possíveis desvantagens da arquitetura SOA pode conter, só será perceptível a partir do momento em que as empresas perceberem que o gerenciamento dos mecanismos de segurança da arquitetura não será uma tarefa fácil. A complexidade na interoperabilidade dos mecanismos de segurança que serão requeridos pelos novos processos de negócios baseados na arquitetura SOA certamente será uma das desvantagens.

A imaturidade dos mecanismos de segurança existentes para aplicações baseadas em XML ou a utilização errônea desses mecanismos associados ao gerenciamento equivocado dos aspectos de segurança (autenticação, autorização, sigilo, não-repúdio etc.) na arquitetura SOA, podem levar o mercado a repensar a implementação SOA no curto a médio prazo como vem sendo previsto.

Em resumo, a imaturidade dos mecanismos de segurança existentes para aplicações baseadas em XML e a preocupação com a complexidade e interoperabilidade

dos mecanismos de segurança na arquitetura SOA nos processos de negócios considerados críticos (principalmente aqueles baseados em Web Services via Internet), podem ser fatores negativos considerados como uma desvantagem da arquitetura.

Capítulo 3

3 Segurança na Arquitetura Orientada a Serviço.

A arquitetura SOA promove através de seus conceitos e regras inovadoras a facilidade de comunicação de forma dinâmica entre serviços interligados por redes como a Internet. Basicamente, essa comunicação ocorre entre serviços (ex. Web Services) provenientes de transações geradas por processos de negócios entre clientes, fornecedores, parceiros de negócios e até mesmo entre serviços mal intencionados. Considerando as facilidades de comunicação e as inovações nos processos de negócios, podemos afirmar que mecanismos de segurança são extremamente necessários para atender aos novos requisitos de segurança que surgem motivados pela arquitetura SOA.

Ray Wagner, diretor de pesquisa do departamento de segurança estratégica do Gatner afirma que o fator determinante da adoção em massa de Web Services depende do sucesso da utilização de tecnologias de segurança baseados em padrões [WSI05]. Considerando que Web Services estão se tornando a plataforma preferida para implementação da arquitetura SOA, podemos imaginar o quanto é importante os mecanismos de segurança atuais e os novos que surgiram.

3.1 A segurança para arquitetura SOA é complexa.

É correto afirmar que a segurança para arquitetura SOA é complexa, e que também a maior parte das empresas candidatas a implementarem a arquitetura SOA não tem a percepção ou o entendimento básico das principais diferenças e desafios dos aspectos de segurança existentes na arquitetura. Naturalmente que com todas as vantagens e benéficos que as empresas esperam alcançar implementando a arquitetura SOA, a preocupação com a segurança esta sendo deixada

para segundo plano mesmo neste cenário de negócio inovador proporcionado pela arquitetura SOA.

Quando os conceitos da arquitetura SOA forem utilizados para prover serviços baseados em Web Services via Internet utilizando os protocolos padrões abertos disponíveis (XML, SOAP, WSDL, UDDI etc.), certamente será necessário também a utilização dos novos mecanismos de segurança. Por exemplo, utilização de mecanismos de segurança focados nas mensagens associados aos mecanismos já existentes para a camada de transporte, serão extremamente importantes para atender os novos requisitos de segurança que a arquitetura SOA poderá proporcionar.

A segurança para negócios baseados em Web Services pode ser implementada em dois diferentes níveis como mencionado anteriormente, mecanismos de segurança para o nível de transporte e mecanismos de segurança voltados para as mensagens. Os mecanismos de segurança utilizados no transporte e nas mensagens, podem ser implementados separadamente ou em conjunto. Cada um desses mecanismos de segurança dispõe de padrões e protocolos específicos para atender os diferentes requisitos de seguranças de cada negócio proporcionados pela arquitetura SOA.

3.1.1 Segurança no Transporte.

Atualmente existe uma grande variedade de serviços eletrônicos oferecidos via Internet, muitos desses serviços são nossos velhos conhecidos e não saberíamos viver sem eles. Estamos falando de serviços bancários oferecidos via Web como, lojas virtuais on-line, inúmeros servidores de web-mail etc. Praticamente, esses serviços já fazem parte de nossa rotina diária o que nos faz bastante confortáveis em utilizá-los sem o menor problema, embora exista uma forte preocupação relacionada à segurança da informação.

O que a grande maioria dos usuários desses serviços não conhece ou acha muito técnico o assunto para entender mesmo os utilizando no dia a dia, são os inúmeros mecanismos de segurança existentes e utilizados nesses serviços. O mecanismo de segurança para esses tipos de

serviços eletrônicos que já estamos acostumados a utilizar, garante o total sigilo dos dados na comunicação entre as nossas máquinas e os servidores que estamos acessando. O mesmo mecanismo de segurança é utilizado também quando empresas efetuam transações eletrônicas com seus parceiros de negócios via EAI, por exemplo.

Os mecanismos e protocolos utilizados para garantir segurança dos dados trocados entre empresas e parceiros, são praticamente os mesmos utilizados nas transações eletrônicas que efetuamos quando compramos um livro ou pagamos uma conta on-line na Internet. Geralmente, as empresas por terem pessoas preparadas para utilizar recursos de segurança adicionais, acabam exigindo mais nos requisitos de segurança nos processos de negócios efetuados eletronicamente. Por exemplo, as empresas que utilizam EAIs podem simplesmente adicionar a autenticação mútua dos provedores e consumidores de serviço utilizando certificados digitais. Além disso também existe a possibilidade de utilizar assinatura digital nos processos bem como cifrar os dados etc.

Todos esses cenários de comunicações que mencionamos estão presentes no nosso dia a dia e cada vez mais se percebe um crescimento dos adeptos. Tudo isso em função do baixo custo de comunicação que a Internet proporciona e o alto nível de segurança que atingimos ao longo do tempo. Os mecanismos responsáveis pela segurança e sigilo dos dados proporcionado nas nossas comunicações na maioria dos casos são alcançados através da utilização dos protocolos SSL (Security Socker Layer) e TLS (Transport Layer Security). Esses protocolos se tornaram padrão de fato do mercado e estão presente na grande maioria das transações seguras efetuadas atualmente na Internet.

Os protocolos SSL/TLS serão abordados seção 3.3.2 e 3.3.3, porém é extremamente importante comentarmos um pouco da característica do SSL/TLS de prover segurança e sigilo dos dados somente para comunicação ponto a ponto. Entendendo melhor a maneira que os protocolos SSL/TLS oferecem segurança, nos ajudará a entender as diferenças entre segurança no transporte e segurança da mensagem e portanto embasar a tese da complexidade de interoperabilidade existente na arquitetura SOA. .

Essa característica de prover somente segurança e sigilo dos dados na comunicação ponto a ponto, é ideal para proporcionar segurança dos serviços que estamos acostumados a consumir, por exemplo, Web e-mail, banco on-line etc. Isso significa que a comunicação entre o consumidor de serviço e o provedor de serviço estará totalmente segura contra violação de sigilo. Obviamente, o modelo de negócio ou os serviços que são consumidos tem como requisito utilizar mecanismos de segurança para garantir segurança e sigilo entre o consumidor e provedor de serviços apenas. E esse requisito tem sido atingido satisfatoriamente com a utilização dos protocolos SSL/TLS.

No entanto, existem novos modelos de negócios proporcionando uma demanda crescente de serviços intermediários entre o consumidor e o provedor de serviços. Apesar da crescente demanda na utilização de serviços intermediários, notamos que os requisitos e os mecanismos de segurança permaneceram praticamente os mesmos. Isso significa que muitos modelos de negócios mesmo que utilizando serviços intermediários no processo de negócio continuam a utilizar mecanismo de segurança como SSL/TLS para garantir a segurança da comunicação, não se importando (talvez pela dificuldade de implementação) com os novos requisitos de segurança que o modelo utilizando interfaces intermediárias proporciona.

Exemplo, quando utilizamos mecanismo de segurança no transporte proporcionado pelos protocolos SSL/TLS, estamos garantindo apenas que a comunicação entre dois serviços seja segura e com total sigilo. Isso ocorre porque a comunicação é estabelecida e cifrada na camada de transporte apenas, o que significa de maneira bem simples receber os dados enviados cifrados na camada de transporte, efetuar o desciframento dos dados e envia-los para a camada de aplicação processar os dados (a explicação das 7 camadas do modelos OSI não faz parte desta dissertação).

Em um processo de comunicação ponto a ponto utilizando os protocolos SSL/TLS, a segurança e o sigilo é totalmente garantida, porém se existim interfaces intermediárias entre provedor e consumidor, o processo de cifrar e descifrar na camada de transporte ira ocorrer toda vêz que os dados trafegarem por um serviço intermediário, fazendo com que o sigilo e a segurança sejam quebrados em cada serviço intermediário. A Figura 04 ilustra um cenário de comunicação utilizando serviços intermediários. Percebemos que o mecanismo de segurança na

camada de transporte não proporciona sigilo para os dados uma vez que os mesmos são recebidos pelo serviço intermediário.

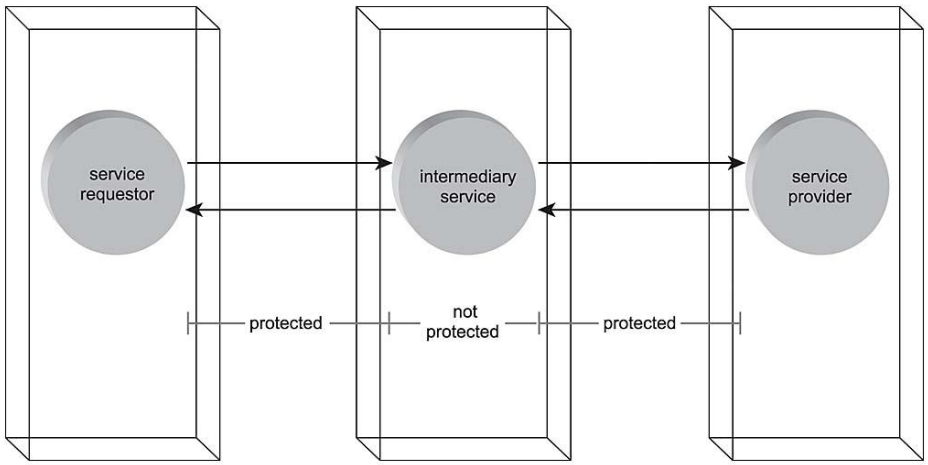


Figura 04 - Transport Level Security, [ERL05]

Comentário.

Proporcionar segurança adequada na camada de transporte quando se utiliza serviços intermediários pode ser uma tarefa complexa. Quem realmente determina a complexidade da adoção de mecanismos de segurança para os serviços providos é o modelo de negócio adotado. São inúmeras alternativas de mecanismos de segurança voltados a proporcionar segurança para todos os tipos de casos simples ou complexos, mas a viabilidade de uso depende da facilidade de gerenciamento, da complexidade da solução, da abrangência e principalmente da interoperabilidade e custo baixo.

Os serviços intermediários estão se tornado parte nos novos processos de negócios proporcionados principalmente pela arquitetura SOA. A arquitetura SOA tem essa característica de facilitar a utilização de serviços intermediários para um determinado processo de negócio. Com essa facilidade surgiu também a dificuldade de prover segurança e sigilo na comunicação quando utilizados serviços intermediários. Os protocolos SSL/TLS, como já foram mencionados anteriormente, não atendem esses novos requisitos de segurança exigidos pela arquitetura SOA,

para isso sugiram novos protocolos para garantir a segurança das mensagens para complementar os mecanismos de segurança já oferecidos pelos protocolos SSL/TLS.

3.1.2 Segurança de Mensagens.

Embora segurança de mensagem não seja muito difundida ou utilizada nos processos de negócios, ela não é algo recente, pelo contrário. A preocupação em prover este nível de segurança surgiu em paralelo com a preocupação de prover segurança do transporte. Porém, com a crescente adoção da arquitetura SOA e principalmente pelas características inovadoras de negócios proporcionados por ela, fez com que segurança de mensagens se tornasse relevante. Uma das características da arquitetura SOA que proporciona o requisito de utilização de mecanismos de segurança de mensagens, é a possibilidade de utilização de serviços intermediários no processo de troca de dados via mensagens SOAP/XML.

Como já sabemos, a arquitetura SOA é baseada na troca de dados através de mensagens SOAP/XML. Essas mensagens geralmente trafegam entre o serviço consumidor e o serviço provedor. Como já observamos também anteriormente, que a arquitetura SOA possibilita a utilização de serviços intermediários que compõe o processo de negócio. Embora o serviço intermediário faça parte do processo de negócio, as vezes ele não necessita ou não deve obter acesso ao dados trocados pelas interfaces da extremidades. Por exemplo, um processo de pagamento eletrônico por cartão de crédito utilizando a arquitetura SOA, poderia ser efetuado sem a necessidade de expor dados sigilosos do cartão de crédito para um serviço intermediário.

Comentário

Atualmente é muito comum fazermos o pagamento eletrônico de compras efetuadas on-line na Internet utilizando cartão de crédito. No processo de pagamento eletrônico geralmente temos uma página Web onde digitamos os dados confidenciais de cartão de crédito para serem validados junto a administradora do cartão, que consequentemente recebe os dados processa os mesmos e efetua a aprovação/recusa do

processo de pagamento da compra. Certamente ficamos preocupamos com a segurança neste tipo de comunicação, no entanto quase a maioria desse tipo de processo de pagamento on-line utiliza os protocolos SSL/TLS para garantir que os dados enviados estejam seguros quando trafegados pela Internet.

O interessante é que pouca gente sabe que os dados confidenciais do cartão de crédito são geralmente expostos no processo de compra on-line, e que essa exposição de dados poderia ser evitada simplesmente deixando de passar em texto aberto pelo serviço intermediário (loja virtual). Isso ocorre em função da particularidade dos protocolos de segurança utilizados na camada de transporte, SSL/TLS, ou seja, os dados enviados são cifrados no serviço consumidor e descifrados no serviço intermediário loja virtual. Depois eles são novamente criptografados e enviados a outro serviço disponibilizado pela administradora do cartão de crédito para validação e aprovação do pagamento da compra.

Observando o processo de pagamento eletrônico em uma transação de compra on-line, chegamos a conclusão que a exposição desnecessária dos dados confidenciais do cartão de crédito poderia ser evitada. Os dados do cartão de crédito é exclusivamente de propriedade do usuário e da administradora do cartão, e qualquer exposição desnecessária desses dados (ex. número de cartão de crédito) sigilosos só aumentaria a possibilidade de quebra na segurança e sigilo.

Contudo, essa situação de expor dados sigilosos como informações de cartão de crédito etc., já pode ser evitada com a utilização do protocolo SET Secure Electronic Transaction. O protocolo SET trabalha na camada de transporte mas oferece funcionalidades ideais para proteger esse tipo de transação eletrônica de pagamento. Porém a utilização e implementação do protocolo SET é considerada muito complexa, essa característica certamente fez com que o protocolo não fosse utilizado em massa nos processos de pagamentos eletrônicos.

Na arquitetura SOA também vamos ter cenários onde serão transacionadas informações confidencias, entre elas dados de cartão de crédito como exemplificamos anteriormente. Como já sabemos os protocolos SSL/TLS não oferecem mecanismos de segurança suficientes para garantir

sigilo caso existam serviços intermediários. O protocolo SET poderia ser uma alternativa, porém além de sua complexidade de utilização e implementação, ele foi concebido para trabalhar na camada de transporte e não iria satisfazer os requisitos de segurança dos novos modelos de negócios que a arquitetura SOA pode proporcionar.

A alternativa de segurança para a arquitetura SOA é recente e nasceu com a demanda crescente da utilização do XML. Com uma arquitetura formada pela troca de mensagens SOAP/XML, se fez necessário a criação de mecanismos de segurança baseados em XML que possibilitasse a segurança e sigilo dos dados trocados. Assim nasceu XML Digital Signature e XML encryption, protocolos que garantem a segurança e sigilo das mensagens. Esses protocolos são basicamente um complemento dos protocolos SSL/TLS, e eles atendem aos exigentes requisitos de segurança proporcionados pela arquitetura SOA.

A Figura 05 abaixo ilustra um cenário onde se utilizam serviços intermediários, e a segurança e sigilo são garantidos em todo caminho entre os serviços das extremidades. A segurança e sigilo são proporcionados pelos protocolos XML Digital Signature e XML encryption que serão discutidos no item 3.3.5 e 3.3.6. Assim como o protocolo SET que mencionamos acima, a utilização e implementação do XML Digital Signature e XML Encryption pode ser complexa dependendo do tipo de negócio e será um desafio para arquitetura SOA. Mas essa discussão referente a complexidade e desafio é o nosso próximo assunto.

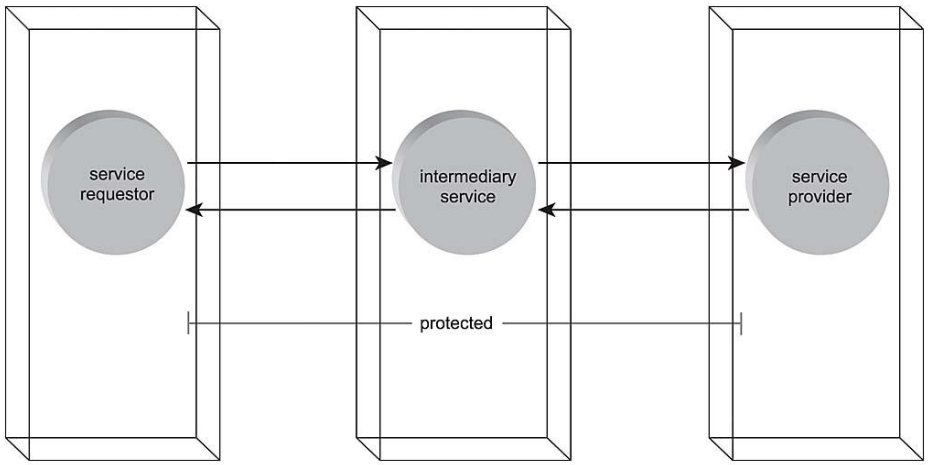


Figura 05 - End-to-End Message Level Security, [ERL05]

3.2 Segurança, um desafio a Arquitetura Orientada a Serviços.

Segurança da informação é um requisito básico para qualquer sistema, aplicação, arquitetura etc., e se torna um grande desafio de implementação, manutenção e gerenciamento em função da complexidade dos requisitos de segurança exigidos pelo negócio. Segurança também pode ser considerada como requisito mandatório quando o processo de negócio disponibiliza Web Services através da Internet. Já a arquitetura SOA faz com que o requisito por segurança seja algo ainda mais desafiador se considerarmos as questões de complexidades e interoperabilidades existentes que iremos identificar a seguir.

Para um melhor entendimento dos desafios que julgamos relevantes na arquitetura SOA, é interessante fazer uma análise dos mecanismos de segurança que as empresas utilizam atualmente nos processos de negócios como, Web (negócios on-line) e arquiteturas utilizando EAIs (ponto-a-ponto) etc.

A disponibilização de serviços através da Web é um modelo de negócio que temos mais familiaridade atualmente em função da grande oferta de serviços oferecidos pela Internet, por exemplo, banco on-line. Os sistemas bancários (banco on-line) no Brasil, citado como exemplo, são conhecidos pela sua interatividade com o usuário cliente; ou seja, o usuário do banco on-line tem a possibilidade de efetuar quase qualquer tipo de transação bancária on-line disponibilizada via Web através da Internet. Certamente, existe uma enorme preocupação com a segurança neste tipo de negócio via Web, e consequentemente cada banco tenta oferecer as melhores práticas e mecanismos de segurança que melhor atendem os requisitos do seu negócio.

Os sistemas bancários (banco on-line) utilizam basicamente mecanismos de segurança na camada de transporte proporcionados pelos protocolos SSL/TLS. Tentando proporcionar ainda mais segurança, alguns bancos estão efetuando testes através de autenticação mútua via certificado digital disponibilizados pelos próprios bancos. Porém, o grande desafio para os bancos em utilizar certificados digitais para autenticação de seus usuários, é encontrar uma forma segura de gerar e distribuir os certificados ao usuário e, ao mesmo tempo, garantir a total interoperabilidade ou mobilidade dos mesmos com o sistema.

Comentário

É um fato concreto que o aumento da utilização de certificado digital nos processos de negócios vem se mantendo principalmente em função dos benefícios de segurança proporcionados por ela. Alguns bancos tiveram a iniciativa de distribuir certificados digitais para seus clientes com o objetivo de aumentar a segurança no processo de autenticação. Em conseqüência disso aqueles clientes que optaram em utilizar certificado digital para aumentar a segurança no processo de autenticação com o banco, sofreram com problemas de interoperabilidade causada por esse mecanismo.

O problema de interoperabilidade causada pelo certificado digital foi em função do requisito de segurança para os usuários de certificados digitais. O requisito de segurança amarrava o certificado digital do usuário com o acesso aos serviços de banco providos pela Web, não deixando alternativa de acesso a não ser utilizando o certificado digital. O processo funcionou até que o usuário se viu necessitado em efetuar acesso por outra máquina que não continha o certificado digital o que acabou gerando problemas de interoperabilidade.

As mesmas dificuldades são encontradas em ambientes Web Services utilizando EAIs onde o requisito de segurança é efetuar autenticação mútua dos serviços com certificados digitais e às vezes assinar digitalmente as mensagens. Questões de interoperabilidade e gerenciamento entre esses serviços se tornam problemas reais desafiando os arquitetos da infra-estrutura em questão. Para contornar esses desafios de interoperabilidades existentes, as partes envolvidas na comunicação efetuam manualmente o gerenciamento dos certificados digitais necessários para validar a autenticação, assinatura digital etc. Essa intervenção manual acaba enfraquecendo significativamente a segurança da arquitetura e causa um custo elevado de gerenciamento dependendo to tamanho da infra-estrutura utilizada.

Com isso podemos observar que o mercado já utiliza mecanismos de segurança para suprir os mais diversos e exigentes requisitos de segurança. Observamos também, que os mecanismos de segurança mais robustos estão cada vez mais utilizando a Infra-estrutura de

Chaves Públicas (ICP) para proporcionar mecanismos fortes de segurança, por exemplo, autenticação mútua por certificado digital e assinatura digital. Porém, tudo isso torna-se um desafio quando se tenta implementar em larga escala como foi observado anteriormente, existe a preocupação com a geração e distribuição dos certificados, a interoperabilidade com a infra-estrutura ICP e o alto custo de administração desses mecanismos de segurança.

Na arquitetura SOA esses desafios de utilização dos mecanismos de segurança que observamos acima ainda continuam existindo, porém a arquitetura SOA acrescenta outros desafios relativamente novos. Por exemplo, os mecanismos de segurança XML digital signature e XML encryption serão utilizados para prover sigilo e segurança na troca de mensagens SOAP/XML entre serviços. De alguma forma esses protocolo terão que estar interoperando dinamicamente com a infra-estrututra de ICP, caso contrário não fará sentido oferecer serviços através de interface endereçável e auto-descritiva proporcionado pela arquitetura SOA.

Recentes padrões e especificações surgiram para abordar esses desafios de integração e interoperabilidade entre serviços SOA e a infra-estrutura ICP. Porém, essa questão não descaracteriza o fato que a Arquitetura SOA enfrentará no curto e médio prazo desafios reais para disponibilizar maneiras seguras de integração com a infra-estrutura ICP para atender os novos requisitos dos processos de negócios baseados em serviços SOA.

Implementação e gerenciamento de mecanismos de segurança não integrados com o ICP, proporciona custos elevados de gerenciamento fazendo com que a arquitetura SOA fique inviável no curto e médio prazo para empresas interessadas nessa arquitetura. Além disso, existe o alto risco com relação a segurança da infra-estrutura de negócio uma vez que a falta de interoperabilidade entre os mecanismos de segurança forçam as empresas a intervirem manualmente em suas configurações, levando ao enfraquecimento da estratégia de segurança adotada.

3.2.1 Garantir a interoperabilidade.

Interoperabilidade dos mecanismos de segurança sem dúvida nenhuma é o grande vilão dos inúmeros desafios que a arquitetura SOA pode enfrentar. Porém, já antecipando tais desafios de interoperabilidade, a Web Services Interoperability Organization (WS-I) [WSI06] que é uma organização sem fins lucrativos e que promove interoperabilidade de Web Services nas plataformas, sistemas operacionais e linguagens, em conjunto com representantes das empresas de software membros da organização, formou grupos de trabalhos com o objetivo de resolver tais questões de interoperabilidade dentre eles mecanismos de segurança.

Esses grupos de trabalhos que foram formados para relacionar e discutir questões relevantes a interoperabilidade de Web Services, bem como desenvolver documentos conhecidos como “profiles” para facilitar o entendimento e discussão das inúmeras especificações disponibilizadas por diferentes órgãos como “Organization for the Advancement of Structured Information Standards” (OASIS) [ORG06], The Internet Engineering Task Force (IETF) [IETF06], World Wide Web Consortium (W3) [W3C06] etc.

Documentos intitulados “profiles” foram disponibilizados ao público para avaliação da comunidade técnica com o objetivo de receber sugestões, críticas etc., visando buscar o máximo de aperfeiçoamento e abrangência do “profile” antes que a versão final oficial seja disponibilizada. O propósito desses documentos elaborados pela WS-I e membros é reunir pontos importantes das especificações (Open Standard), esclarecer, aportar correções e servir como um material guia de referência para assuntos de Web Services etc.

A WS-I define, “Profile disponibiliza um guia de implementação para facilitar a utilização e interoperabilidade entre especificações Web Services.” A WS-I também menciona que recentemente finalizou Basic Profile, Attachments Profile e Simple SOAP Binding Profile, mas que ainda existe um trabalho para finalizar o Basic Security Profile [WSD05]. Algumas das principais profiles eleboradas pela WS-I estão relacionadas abaixo como referência, Lista de Profiles (Basic Profiles Working Group) –[lista 02] e Lista de Profiles (Basic Security Profile

Working Group) - lista [03], informações mais detalhadas de cada uma delas pode ser encontrada na página oficial da WS-I http://www.ws-i.org.

Lista 02 - Profiles (Basic Profiles Working Group) –WS-I, [DEL05].

- Attachment Profile 1.0
- Basic Profile 1.1
- Simple SOAP Binding Profile 1.0

A documentação "Basic Security Profile" embora ainda não finalizada, aborda assuntos relevantes a interoperabilidade, e também faz inúmeras recomendações nos aspectos de segurança em diversos tópicos tais como: Transport Layer Security, SOAP Message Security, Username Token Profile, X.509 Certificate Token Profile, XML-Signature, XML Encryption, Algorithms, Relationship of Basic Security Extension Profile to Basic Profile, and Attachment Security. O documento incorpora por referência outros assuntos relevante: HTTP over TLS; Web Services Security: SOAP Message Security; Web Services Security: Username Token Profile; Web Services Security: X.509 Token Profile; XML-Signature Syntax and Processing; Web Services Security: SOAP Message Security Section 9; XML Encryption Syntax and Processing.

Lista 03 - Profiles (Basic Security Profile Working Group) – WS-I, [DEL05].

- Basic Security Profile
- Kerberos Token Profile
- REL Token Profile
- SAML Token Profile
- Security Challenges, Threats and Countermeasures

Todos os "profiles" desenvolvidos pela organização WS-I fazem referências a outros "profiles" desenvolvidos pela própria WS-I e especificações desenvolvidas pelas organizações OASIS, IETF, W3C etc. que também são organizações internacionais sem fins lucrativos que tem como objetivo direcionar desenvolvimentos de padrões Web Services em geral. Como exemplo, segue a lista de alguns "profiles" e especificações desenvolvidos pela organização OASIS, Lista de Profiles e especificações – OASIS - lista [4]. Para obter lista completa de todas as especificações e "profiles" desenvolvidos pela organização consulte a página oficial da organização [OAS05].

Lista 04 - Profiles e especificações – OASIS, [LIS05].

- UsernameToken Profile 1.0
- Rights Expression Language (REL) Token Profile 1.1
- SOAP Message Security 1.1
- SAML Token Profile 1.1
- SOAP Messages with Attachments (SwA) Profile 1.1
- UsernameToken Profile 1.1
- X.509 Certificate Token Profile 1.1

Embora existam inúmeras especificações e "profiles" elaborados especificamente para tratar assuntos de interoperabilidade entre Web Services, ainda podemos encontrar situações onde questões de segurança relacionadas principalmente a interoperabilidade ainda desafiam a arquitetura SOA. Por exemplo, no item 7 - X.509 Certificate Token Profile da documentação "Basic Security Profile" [PGP06] descreve que Certificate Authority (CA), deve ser uma questão acordada entre ambas as partes que se comunicam.

Esse acordo pode determinar a troca de CA de forma manual entre ambas as partes envolvidas, isso pode tornar inviável uma arquitetura SOA utilizando, por exemplo, no nível de mensagens (XML Digital Signature, XML Encryption) ou serviços que requisitam autenticação mútua por certificados digitais. Lembrando sempre que característica de interfaces endereçáveis e auto-descritivas proporcionado pela arquitetura SOA não faria sentido caso houvesse a necessidade de intervenção manual para troca de certificados.

Comentário

Para exemplificar o desafio que será praticar manualmente a troca de Certificados entre partes envolvidas, vamos imaginar uma arquitetura SOA que utiliza Web Services no seu processo de negócio e tem como requisito mínimo de segurança, assinatura digital nas transações de negócios via XML Digital Signature, bem como, autenticação mútua dos serviços. Supondo também que nesse cenário vamos ter um número ilimitado de possíveis candidatos a consumidores desses serviços. Nessa ilustração fictícia porém realística, certamente haverá a necessidade de gerenciamento enorme de importação de todas as CAs raizes que não são ainda confiadas pelo provedor

de serviço. Caso contrário não existe maneira de validar ou verificar a autenticidade da assinatura digital e autenticação mútua dos inúmeros candidatos a consumidores de serviços.

A situação de interoperabilidade desse cenário ilustrado Figura 06 fica ainda mais desafiador se os possíveis consumidores dos serviços forem provenientes de outros países, apesar das barreiras como distância e linguagem, ainda existe a possibilidade de leis federais impostas por alguns países que podem ser um item do requisito de segurança para se consumir um serviço. O cenário ilustrado Figura 06 é um tipo de negócio que cada vez ganha mais adeptos no mercado, ele é conhecido como leilão reverso.

O leilão reverso atualmente é praticado através de arquitetura Web/Web Services com parceiros pré-determinados, ou seja, as interfaces são totalmente acopladas e a troca de certificados digitais (CAs) raizes são efetuadas de forma manual com antecedência tornando o custo de gerenciamento relativamente alto dependendo do tamanho da solução além do aumento da possibilidade do risco de enfraquecimento da segurança.

Como a tendência do mercado é migrar para arquiteturas mais robustas e inovadoras, por exemplo, arquitetura SOA, podemos imaginar uma situação de leilão eletrônico de carnes bovinas de empresas frigoríficas brasileiras, onde os possíveis clientes poderiam ser inúmeros compradores nacionais e estrangeiros. Em um cenário como este será fácil imaginarmos a arquitetura SOA sendo utilizada para este tipo de negócio, visto que as características proporcionadas por ela facilitam o processo de negócio. Porém, fica a questão de como será a definição dos requisitos de segurança para esse tipo de negócio sem proporcionar barreiras de interoperabilidades.

Se imaginarmos que para garantir a autenticidade e sigilo da comunicação nesse tipo de negócio seria interessante autenticação mútua entre servidores, autenticação mútua dos serviços, cifrar dados relevantes a transações e por fim assinar digitalmente a mensagem, certamente teríamos em nossas mãos um grande desafio para colocar em prática tudo isso. Embora já exista protocolo para tornar possível esse tipo de

interoperabilidade com o infra-estrutura de chaves públicas (ICP), não temos ainda a infra-estrutura montada para estar oferecendo esse serviço em escala mundial.

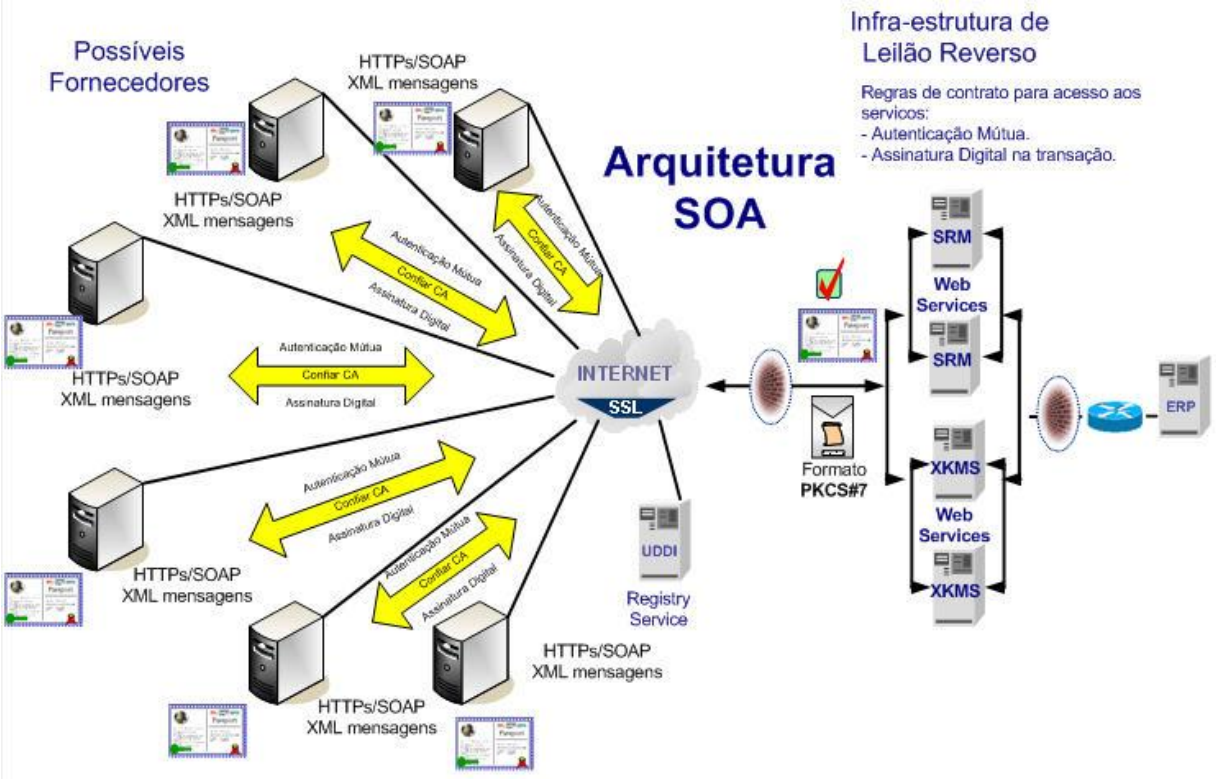


Figura 06 - Arquitetura SOA e o desafio para troca de certificados (CA)

Essa arquitetura SOA ilustrada acima contem inúmeros desafios de segurança conforme segue:

- Gerenciamento de chaves públicas de CAs (Inclusão/Revogação).
- Gerenciamento de chaves públicas dos fornecedores (Revogação).
- Possibilidade de manter chaves privadas e públicas dos usuarios do serviços expostas na DMZ.
- Possibilidade de manter chaves privadas das interfaces de serviços de Registro e Validação exposta na DMZ.
- Gerenciar relação de confiança entre serviço de registro e serviço consumidor.
- Servidor XKMS é ponto único de falha.
- Hardware adicionais para proteção de DMZ (Proxy e firewall).
- Hardware adicional para proteção das chaves privadas.
- Serviços UDDI não esta disponível em larga escala.

O ideal para resolver o desafio de interoperabilidade para troca de Certificados Digitais públicos, seria utilizar um mecanismo de comunicação para integrar aplicações baseadas em XML com a infra-estrutura de chaves públicas (ICP). Isso iria garantir a interoperabilidade associado ao dinamismo proporcionado pela arquitetura SOA, e evitaria a necessidade de um custo elevado de gerenciamento de Certificados Digitais pelas partes envolvidas.

Na verdade, já existe uma especificação "XML Key Management Specification" (XKMS), que proporciona justamente mecanismos seguros necessários para integrar ICP com aplicações baseadas em XML que necessita utilizar certificados digitais. Isso sem dúvidas cuida da questão de interoperabilidade causado pelo gerenciamento de Certificados, e demonstra que existe uma preocupação sobre o assunto. O desafio é colocá-lo em prática globalmente dentro de um período de curto a médio prazo e resolver algumas questões de segurança desse modelo como lista da na ilustração acima.

Os mecanismos de segurança XML Digital Signature e XML Encryption, dependem necessariamente de uma infra-estrutura ICP (Infra-estrutura de Chaves Públicas) para tornar viável e possível cifrar, descifrar, assinar digitalmente, verificar autenticidade de documentos etc. Para que seja possível utilizar esses mecanismos criptográficos (XML Digital Signature e XML Encryption) de forma dinâmica, será necessário que haja uma integração entre a aplicação XML e uma Infra-estrutura de Chaves Públicas ICP. Isso levando em consideração que as partes que se comunicam estejam utilizando o padrão X.509, a mais popular e bem difundida no momento [VER04].

A seguir temos também a lista das especificações desenvolvidas pelo W3C para tratar assuntos de assinatura digital, criptografia e gerenciamento e chaves relacionados com aplicações XML.

Lista 05 - Especificações W3C para tratar criptografia em XML.

- XML Digital Signature (XMLDSIG)
- XML Encryption (XMLENC)
- XML Key Management Specification (XKMS)

A especificação XML Key Management System 2.0 (XKMS) elaborado pela W3C realmente já esta abordando o desafio de integrar a Infra-estrutura de Chaves Públicas (ICP). O XKMS visa tornar flexível e dinâmico a utilização de aplicações XML com mecanismos de segurança, por exemplo, XML Digital Signature. No entanto, isto não invalida o fato que abordar a questão apenas não significa resolver todos os desafios existentes que a arquitetura SOA enfrenta com relação ao gerenciamento e distribuição de Certificados Digitais. No curto e médio prazo, a arquitetura SOA vai demandar a integração com a Infra-estrutura de Chaves públicas (ICP), e isso pode ser um grande problema se o mercado não estiver pronto para oferecer esta interoperabilidade.

Em resumo, a especificação XKMS 2.0 foi recentemente finalizada pelos órgãos competentes W3C e IETF, só após sua aceitação como especificação Standard de fato começa o trabalho do orgão WS-I em propor a inclusão dessa especificação no Basic Security Profile. Além disso, contamos com o dinamismo do mercado em disponibilizar aplicações Web Services e infra-estrutura baseadas em XML que suporte esta nova Standard XKMS.

Comentário

Atualmente, são poucas a empresas que tem familiaridade com infra-estrutura ICP e a utilizam de forma a prover segurança no nível de transporte e mensagem dentro de seus domínios. A dificuldade, ou melhor dizendo o desafio, será disponibilizar essa infra-estrutura imediatamente para alavancar a utilização da arquitetura SOA na sua introdução inicial, caso contrário a adesão estimada da arquitetura SOA para serviços baseados em Web Services via Internet poderá ser prejudicada mediante as dificuldades existentes com segurança.

3.3 Mecanismos de Segurança.

Atualmente o mercado conta com diferentes tipos de protocolos e mecanismos de segurança que visam proporcionar e garantir segurança e sigilo na troca de informações. Cada um desses

mecanismos tem funcionalidade e aplicabilidade específica de acordo com o tipo de negócio ou requisito de segurança exigido. Por exemplo, os protocolos SSL/TLS juntamente com a Infra-estrutura de Chaves Públicas (ICP) são de fato os mecanismos de segurança mais utilizados mundialmente na Internet para prover segurança na troca de informações de negócios entre empresas, parceiros e clientes (Comercio eletrônico, EAI, Bancos etc.).

Os protocolos SSL/TLS juntamente com a infra-estrutura de ICP proporcionam mecanismos de segurança ponto a ponto. Mecanismo de segurança ponto a ponto garante segurança e sigilo das informações no nível de transporte entre o serviço consumidor e serviço provedor da informação. No entanto, esse mecanismo de segurança proporcionado não atende os requisitos de prover segurança e sigilo para processos de negócios que efetuam troca de informações ou mensagens através de serviços intermediários. Isso porque os protocolos SSL/TLS foram concebidos somente para atuarem como mecanismos de segurança ponto a ponto.

Para atender os novos requisitos de segurança proporcionados pelos novos modelos de negócios e aplicações baseadas em XML, o mercado disponibilizou protocolos e mecanismos de segurança para atuar no nível de mensagem. Isto significa que a segurança e sigilo dos dados trocados entre serviço provedor e serviço consumidor através de serviços intermediários serão garantidos utilizando esses protocolos. Esses protocolos foram desenvolvidos especialmente direcionados para aplicações baseadas em XML com o objetivo de prover, por exemplo, assinatura digital (XML digital signature), criptografia (XML encryption) e interoperabilidade/integração (XML Key Management Specification - XKMS).

Com a estimativa de crescimento do mercado [MT05] utilizando a arquitetura SOA, certamente esses novos mecanismos de segurança serão a peça chave para garantir segurança e sigilo nos processos de negócios baseados na troca de mensagens SOAP/XML. Lembrando que a interoperabilidade entre esses novos mecanismos de segurança ainda é um grande desafio para arquitetura SOA como já foi observado anteriormente no item 3.2.1. Embora os mecanismos de segurança para aplicações baseadas em XML já existam, o mercado agora deve proporcionar rapidamente interoperabilidade entre eles para não frustrar as expectativas de todos que apostam neste crescimento.

3.3.1 Introdução a XML

Extensible Markup Language (XML) foi desenvolvido pela World Wide Web Consortium (W3C) que apresentou sua primeira versão em 1996. W3C utilizou como base de criação da linguagem XML a então popular Standard Generalized Markup Language (SGML - ISO 8879) que existia desde da década de 70 [DEV05]. XML atualmente é uma linguagem utilizada em larga escala pelos processos de negócios voltados principalmente para o ambiente Web Services. A linguagem XML permite a troca de informações simples ou complexas entre empresas, clientes, parceiros, aplicações etc. XML rapidamente está se tornando a linguagem de negócio para transações efetuadas entre sistemas legados, parceiros, empresas, clientes etc.

A grande vantagem da linguagem XML é possibilitar que aplicações escritas em diferentes linguagens, com diferentes estruturas de dados armazenados e diferentes plataformas se comuniquem, ou seja, XML é independente de sistema e plataforma [COU05]. XML provê a representação lógica dos dados, o qual pode se representado em diferentes maneiras. A linguagem XML é flexível e universal e também conta com uma grande quantidade de ferramentas e pacotes para desenvolvimentos disponíveis no mercado. XML ganhou popularidade no final da década de 90 durante o movimento de comercio eletrônico.

Inúmeras aplicações utilizam XML para troca de mensagens aproveitando essa característica de possibilitar a troca de dados entre sistemas e plataformas diferentes. Por exemplo, no mercado brasileiro atualmente, esta havendo uma demanda crescente por aplicações EAIs para troca de mensagens entre empresas parceiras, bancos, sistemas etc. São empresas que efetuam envio de pagamentos e cobranças para os bancos através de mensagens XML entre outras. Toda essa troca de informações como já mencionamos anteriormente efetua proteção das informações através de mecanismos de segurança como SSL/TLS, assinatura digital etc.

Atualmente a linguagem XML é padrão de fato para inúmeros processos de negócios efetuados via Web Services na Internet. Com o surgimento da Arquitetura Orientada a Serviços (SOA) e a perspectiva de sua adoção em larga escala pelo mercado, fez com que a linguagem XML se tornasse ainda mais expressiva no cenário mundial de negócios.

3.3.2 SSL - Secure Sockets Layer

O protocolo SSL foi desenvolvido pela empresa Netscape Communication Corporation e atualmente é o protocolo mais utilizado no mundo para prover segurança e sigilo para processos de negócios disponibilizados principalmente na Internet [PRO06]. Por exemplo, é fato que praticamente toda transação bancária on-line, compra on-line, negócios etc. efetuadas na Internet utiliza o protocolo SSL sobre o protocolo http (https) para prover segurança e sigilo das informações trocadas entre as interfaces. O protocolo SSL tem a finalidade de oferecer sigilo e segurança na comunicação ponto a ponto entre cliente e servidor (entre consumidor de serviços e provedor de serviço) etc.

Basicamente, no processo de estabelecimento de comunicação utilizando o protocolo SSL, o servidor o qual se esta acessando apresenta o certificado digital público padrão X.509v3 para provar a autenticidade de sua identidade e posteriormente estabelecer a comunicação. O cliente (Web Browser) recebe o certificado Digital público do servidor e efetua a verificação e validação do mesmo observando informações como, validade do certificado, endereço domínio, assinatura digital do emissor do certificado etc. para concluir o estabelecimento da comunicação. Dependendo do requisito da comunicação, o servidor pode também requisitar o certificado digital público do cliente para estabelecer a comunicação (autenticação mútua).

Podemos notar que ao utilizar certificados digitais no processo de estabelecimento de comunicação, ambas as partes envolvidas dependem de uma infra-estrutura ICP em comum estabelecida para verificação e validação da autenticidade dos certificados digitais trocados. Isso também não será diferente para o processo de criptografia dos dados utilizando certificado digital e assinatura digital de documentos. Essa dependência por interoperabilidade já é problema atualmente, embora a infra-estrutura de chaves públicas (ICP) seja mundialmente utilizada e difundida. Por exemplo, no Brasil contamos com o ICP-Brasil para reger a nossa infra-estrutura de chaves públicas brasileiras.

A infra-estrutura de chaves públicas (ICP) é fundamental no processo de estabelecimento de comunicação dos protocolos SSL/TLS como foi visto anteriormente. A infra-estrutura ICP

provê informações pertinentes ao certificado digital de forma a garantir a autenticidade do mesmo proporcionando a segurança e sigilo na comunicação. Sem uma infra-estrutura em comum de chaves públicas estabelecida, fica impraticável a utilização de certificados digitais no processo de comunicação via internet utilizando os protocolos (SSL/TLS).

Para evitar problemas ainda maiores com interoperabilidade entre serviços utilizando os protocolos SSL/TLS, empresas de software como Microsoft e Netscape já disponibilizam em suas aplicações as principais CAs raízes existentes no mercado. Isso resolve parte do problema, porém existem inúmeras CAs raízes que não são disponibilizados por essas empresas, além disso existem aquelas CAs que foram criadas recentemente ou ainda aquelas que foram disponibilizadas e expiraram a validade. A conclusão de tudo isso é que o mecanismo de distribuição e atualização de CAs raízes não é feita de forma integrada, e ainda não estamos mencionamos os mecanismos de revogação dos certificados digitais.

Nos processos de negócios baseados em aplicações XML e na arquitetura SOA, os requisitos por segurança e sigilo são praticamente os mesmos diferenciando apenas nos mecanismos de segurança utilizados. Por exemplo, quando utilizamos os protocolos SSL/TLS estamos garantindo segurança e sigilo ponto a ponto no nível de transporte entre o provedor e consumidor de serviços. Na arquitetura SOA isso tende a ser diferente, aplicações baseadas em XML trocarão mensagens SOAP/XML que poderão trafegar por vários serviços intermediários ate chegar ao seu destino final. Desta forma, utilizar apenas os protocolos SSL/TLS tornaria impossível a garantia de segurança e sigilo das informações entre os serviços.

Para resolver esta questão surgiram novos mecanismos de segurança baseados em XML focando prover segurança e sigilo nas mensagens XML trocadas entre aplicações baseadas na arquitetura SOA. Com esses novos mecanismos é possível garantir segurança e sigilo das informações mesmo que o processo de negócio utilize serviços intermediários. Tudo isso é possível em função dos padrões XML digital signature e XML encryption que tratam dessa questão tornando viável a garantia de segurança e sigilo nas trocas de mensagens SOAP/XML.

Assim como o protocolo SSL necessita interoperabilidade com infra-estrutura ICP os novos mecanismos de segurança baseados em XML também necessitam. Para garantir a interoperabilidade que surgiu entre a necessidade de integrar a infra-estrutura de ICP existente com aplicações baseadas em XML, foi criado o protocolo XML Key Management Service XKMS. A função do XKMS é integrar a infra-estrutura ICP com aplicações baseadas em XML, tornando possível para processos de negócios baseados na arquitetura SOA validarem dinamicamente os certificados digitais que serão utilizados juntamente com o processo de assinatura digital, criptografia etc.

3.3.3 TLS - Transport Layer Security.

O protocolo TLS desenvolvido pela Internet Engineering Task Force (IETF) é o sucessor do protocolo SSLv3 desenvolvido pela empresa Netscape Communication Corporation. O TLS foi baseado no protocolo SSLv3 e mantém praticamente as mesmas funcionalidades do protocolo SSLv3. O protocolo TLS tem vantagens e desvantagens com relação ao seu antecessor, mas o importante aqui é frisar que o protocolo TLS também oferece os mesmos mecanismos de segurança e sigilo oferecidos pelo protocolo SSLv3. Versões mais recentes de Web Browsers e Servidores Web já oferecem suporte ao protocolo TLS.

3.3.4 ICP – Infra-estrutura de Chaves Públicas.

“A criptografia é uma ciência que tem importância fundamental para a segurança da informação, ao servir de base para diversas tecnologias e protocolos, tais como a infra-estrutura de chaves públicas (ICP), o IP Security (IPSec e o Wired Equivalent Privacy (WEP)”. [NG04]. “O termo Public Key Infrastructure (PKI) ou ICP é usado para descrever processos, políticas e Padrões que governam a emissão, manutenção e revogação dos certificados públicos e privados que operações de criptografia e assinatura digital requisitam” [SMA01].

Atualmente, os principais mecanismos de segurança e sigilo disponíveis para transações eletrônicas utilizam certificados digitais proporcionados pela infra-estrutura ICP para prover segurança nas transações de negócios efetuadas na Internet. Os protocolos SSL/TLS são os que mais utilizam a infra-estrutura ICP para proporcionar segurança e sigilo para essas transações de negócios. Como exemplo, podemos citar alguns segmentos de negócios como bancos, comércio eletrônico e aplicações EAI etc.

Comentário

O mercado brasileiro tem mostrado uma tendência na utilização de certificados digitais para autenticação mútua bem como assinatura digital nas transações eletrônicas efetuadas na Internet. Autenticação mútua e assinatura digital vem reforçar os mecanismos de segurança proporcionados pelos protocolos SSL/TLS que vem sendo utilizados fortemente desde a década de 90. Os bancos principalmente se mostraram bastantes receptivos em adoção de assinatura digital nas transações eletrônicas efetuadas pelos seus clientes jurídicos a princípio. Por exemplo, o envio de partidas de pagamentos e cobranças, folha de pagamento e download de extratos, estão cada vez mais utilizando mecanismos criptográficos tipo assinatura digital para proporcionar mais segurança nos processos de negócios efetuados eletronicamente.

A utilização da assinatura digital em transações efetuadas eletronicamente garante a autenticidade das informações e consequentemente o não repúdio. Para que o processo de assinatura digital nas transações eletrônicas entre bancos e empresas seja possível e esteja embasada nas leis brasileiras, ambas as partes devem conter um par de chaves criptográficas (pública e privada). As chaves devem estar assinadas por uma Autoridade Certificadora (CA) reconhecida pela infra-estrutura de ICP brasileira ICP-Brasil conforme Medida provisória MP 2.200-2 que Institui a Infra-Estrutura de Chaves Públicas Brasileira - ICP-Brasil [l3].

A infra-estrutura de ICP pode também proporcionar base para mecanismos de segurança como, por exemplo, Single sign-on (SSO), Identity Federation etc. Muitas empresas estão

optando em estabelecer ICP privada dentro de seus domínios justamente para oferecer embasamento para utilizar certificado digital para proporcionar mecanismos de SSO para suas aplicações. O funcionamento de uma infra-estrutura ICP privada é praticamente o mesmo da infra-estrutura ICP pública. A ICP privada oferece mais flexibilidade de uso e reduz custo de investimento em certificados digitais, porém manter a segurança de uma infra-estrutura privada não é tarefa simples.

Muitas empresas que optaram por manter uma infra-estrutura privada de ICP se beneficiaram com a robustez nos aspectos de segurança oferecidos pela infra-estrutura e a redução significativa do custo de manutenção de gerenciamento de usuários (autorização de acesso). Por outro lado, quando essa infra-estrutura de ICP privada é também utilizada para oferecer segurança entre parceiros, clientes etc., observou-se a complexidade de implementação e o alto custo de gerenciamento necessário para esse mecanismo de segurança. Isso ocorre necessariamente pelo fato de não existir um mecanismo que ofereça integração dinâmica para facilitar a interoperabilidade dos certificados digitais das CA raízes, por exemplo.

Hoje podemos apontar que o principal desafio e o principal inibidor da utilização em massa dos certificados digitais em processos de negócios efetuados eletronicamente, é sem dúvida a complexidade de implementação desses mecanismos de segurança entre diferentes domínios. A geração e distribuição dos certificados e o estabelecimento da relação de confiança entre domínios exigidos pelo mecanismo de segurança é atualmente proporcionados através de processos manuais. Esses desafios atuais tendem a permanecer na arquitetura SOA também, e certamente irão somar aos novos desafios que serão proporcionados pelos requisitos de segurança dos novos processos de negócios baseados em XML.

Na arquitetura SOA os requisitos por mecanismos de segurança é semelhante aos atuais como já foi observado anteriormente, porém os processos de negócios baseados em XML proporcionarão na arquitetura SOA requerem segurança e sigilo no nível de mensagem. Desta forma cifrar ou assinar digitalmente a mensagem XML ou parte dela, as vezes se faz necessário dependendo do requisito de segurança adotado para cada negócio.

Outro exemplo de negócio que vai utilizar aplicações XML bem como os mecanismos de segurança como assinatura digital e relógio de tempo é a nota fiscal eletrônica (NFe). A NFe é um projeto dos governos de vários estados (BA, GO, MA, SC, SP e RS) que propõe a criação de notas fiscais eletrônicas para diminuir os custos com formulários padronizados, reduzir a falsificação e notas e sonegação de impostos. Todo o processo de NFe vai estar utilizando a linguagem XML, certificado digitais no processo de autenticação e assinatura. O projeto é audacioso e os aspectos de segurança certamente será um desafio [NFE06].

Os mecanismos de segurança baseados em aplicações XML para arquitetura SOA como assinatura digital e criptografia para XML, autenticação mútua dos serviços, servidores, e usuários etc., dependerá necessariamente de uma integração dinâmica com a infra-estrutura de ICP já existente para proporcionar a interoperabilidade. Atualmente não existe nenhum serviço global disponível para efetuar esta integração, embora já existam os protocolos para tornar isso possível (XKMS, XKISS e XKRSS). Uma integração entre ICP e aplicações baseadas em XML é um grande desafio e extremamente necessário, e sem dúvidas o requisito fundamental para não inibir e prorrogar a estimativa de adesão da arquitetura SOA pelo mercado [MT05].

3.3.5 XML Digital Signature.

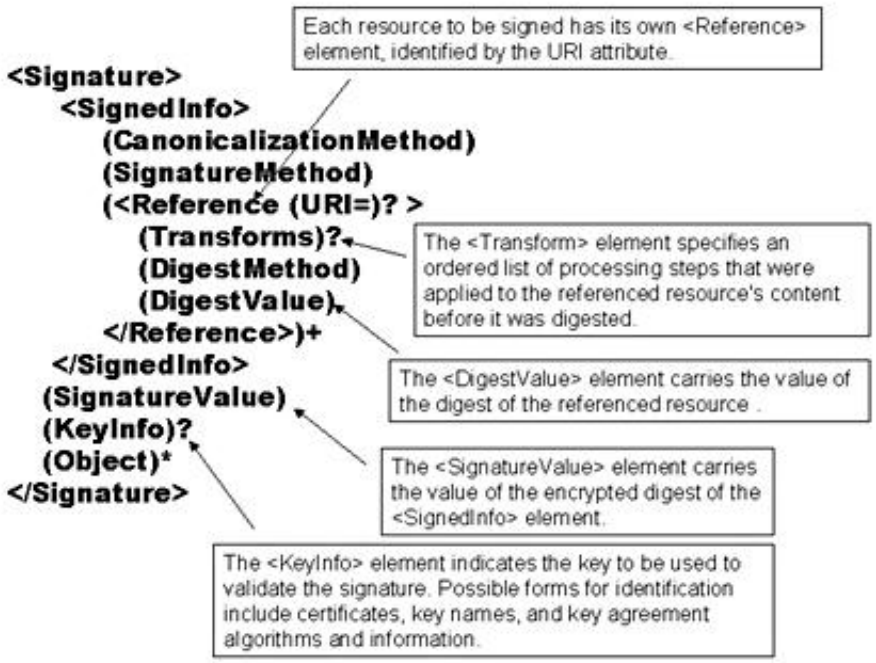
A Standard XML digital signature foi desenvolvida pela parceria das organizações World Wide Web Consortium (W3C) e Internet Engineering Task Force (IETF). A Standard proporciona regras e sintaxes para criar e representar assinatura digital para transações baseadas em XML garantindo o não repúdio, integridade, autenticação da mensagem e ou autenticação do autor [XML02]. Uma das características fundamentais do XML digital signature é a habilidade e flexibilidade de assinar digitalmente somente parte da mensagem XML [SMA01]. Essa flexibilidade é muito importante para os novos modelos de negócios baseados nas aplicações em XML e principalmente para arquitetura SOA.

XML digital signature como mencionado anteriormente depende necessariamente de uma infra-estrutura de ICP para viabilizar o processo de não repúdio, integridade e autenticação. A integração com uma infra-estrutura de ICP é requisito fundamental para o processo de verificação

e validação da assinatura digital. Os modelos de negócios atuais baseados em Web Services ou mesmo em aplicações XML baseadas em EAIs, utilizam assinatura digital baseados nos padrões PKCS#7 ou CMS. Os novos modelos de negócios baseados na arquitetura SOA utilizam o padrão XMLDsig para garantir os requisitos de segurança conforme exigência do negócio.

A seguir são apresentados os componentes de uma assinatura digital baseada em XML. Informações referente XML Digital signature, bem como todos os algoritmos criptográficos recomendados e suportados pela Standard podem ser encontrados na página oficial da W3C http://www.w3.org/TR/xmldsig-core/.

Lista 06 - Componentes de uma assinatura XML, [SMA01].



3.3.6 XML Encryption.

XML Encryption Standard foi desenvolvida pela organização World Wide Web Consortium (W3C) formado por um grupo de trabalho composto por varias empresas como, Microsoft, Motorola, IBM, Sun Microsystems, VeriSign , BEA Systems, entre outros. A Standard XML Encryption proporciona regras e sintaxes para criar e representar criptografia para garantir o sigilo em transações baseadas em XML. Uma das características fundamentais do XML Encryption é a habilidade e flexibilidade de criptográfar somente parte da mensagem XML [ENC05]. Essa flexibilidade é muito importante para os novos modelos de negócios baseados nas aplicações XML e principalmente para arquitetura SOA.

Existem vários cenários de negócios baseados em aplicações XML e não XML que dependem de mecanismos de segurança para garantir sigilo dos dados durante a comunicação. Por exemplo, transações de compras via Internet onde dados confidencias são transmitidos entre cliente, loja on-line e administradora de cartão de crédito. Neste caso temos mais de um serviço receptor de dados para um cliente em comum, ou seja, no processo de envio de numero de cartão de crédito para efetivar uma compra on-line, a loja on-line não necessita conhecer o numero do cartão de crédito do cliente, basta simplesmente ela pegar a aprovação da operadora do cartão para efetivar a venda ao cliente.

Para esse tipo de transação de negócio de validação de cartão de crédito baseado em aplicações XML, poderíamos simplesmente utilizar o mecanismo de criptografia para XML, ou seja, os dados pertinentes ao cartão de crédito (número do cartão e vencimento) ficariam criptografados entre o cliente e a operadora do cartão. Esse mecanismo de segurança proporcionado pelo XML encryption, beneficia a loja on-line que não necessita se preocupar com a segurança dos dados confidenciais como numero do cartão de crédito, e a operadora juntamente com o cliente visto que as informações do cartão crédito trocados permanece em sigilo total.

Comentário.

Sabemos que protocolos como SSL/TLS também garantem sigilo na comunicação, porém o sigilo é provido somente para comunicação ponto a ponto e no nível de transporte, algo que não atende os novos requisitos de aplicações XML. No exemplo acima, os protocolos SSL/TLS e SET não atenderiam os requisitos de segurança proporcionados pelo modelo de negócio baseados em aplicações XML. A segurança e o sigilo requeridos pelo processo de negócio somente poderia ser alcançados com a utilização do XML encryption uma vez que existem serviços intermediários entre os serviços provedor e consumidor.

É importante ressaltar que XML encryption não é de maneira nenhuma um substituto ou alternativa aos protocolos SSL/TLS ou SET. A Standard XML encryption juntamente com o XML digital signature foram desenvolvidos para disponibilizar mecanismos de segurança não proporcionados pelos protocolos SSL/TLS e SET. A crescente utilização do XML para aplicações Web e grande expectativa de adoção pelo mercado da arquitetura SOA, certamente contribuiu para o desenvolvimento da Standard XML encryption. O resultado é que mercado agora conta com mais um mecanismo de segurança para prover segurança e complementar os requisitos gerados pelos novos processos de negócios baseados em XML.

O grande desafio de utilizar XML encryption em um modelo de negócio como esse que foi mencionado acima, é garantir total interoperabilidade. Para que seja possível aplicação do cliente (Web Browser) efetue a criptografia dos dados confidenciais como numero do cartão de crédito, a aplicação deveria contemplar essa funcionalidade. Além disso existe a relação de confiança, ou seja, tem que existir em algum momento o envio da chave pública para que seja efetuada a criptografia baseada nela.

Para exemplificar algumas das características importantes que a Standard XML encryption vai prover, esta sendo demonstrado logo abaixo exemplos de códigos XML que serão criptografados de diversas formas (o arquivo XML inteiro, um elemento do XML e o conteúdo de um elemento).

Um exemplo de arquivo XML a ser encriptado, [SID02]

```
<purchaseOrder>
        <Order>
                <Item>book</Item>
                <Id>123-958-74598</Id>
                <Quantity>12</Quantity>
        </Order>
        <Payment>
                <CardId>123654-8988889-9996874</CardId>
                <CardName>visa</CardName>
                <ValidDate>12-10-2004</ValidDate>
        </Payment>
</purchaseOrder>
```

Encriptando o arquito inteiro. [SID02]

```
<?xml version='1.0' ?>
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
                Type='http://www.isi.edu/in-notes/iana/assignments/media-
types/text/xml'>
        <CipherData>
                <CipherValue>A23B45C56</CipherValue>
        </CipherData>
</EncryptedData>
```

Encriptando somente o elemento <Payment>, [SID02]

```
<?xml version='1.0' ?>
<PurchaseOrder>
        <Order>
                <Item>book</Item>
                <Id>123-958-74598</Id>
                <Quantity>12</Quantity>
        </Order>
        <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'

xmlns='http://www.w3.org/2001/04/xmlenc#'>
                <CipherData>
                        <CipherValue>A23B45C564587</CipherValue>
                </CipherData>
```

```
            </EncryptedData>
        </PurchaseOrder>
```

Encriptando somente o conteudo dentro do elemento <CardId>, [SID02]

```
        <?xml version='1.0' ?>
        <PurchaseOrder>
            <Order>
                <Item>book</Item>
                <Id>123-958-74598</Id>
                <Quantity>12</Quantity>
            </Order>
            <Payment>
                <CardId>
                    <EncryptedData
Type='http://www.w3.org/2001/04/xmlenc#Content'

        xmlns='http://www.w3.org/2001/04/xmlenc#'>
                        <CipherData>
                            <CipherValue>A23B45C564587</CipherValue>
                        </CipherData>
                    </EncryptedData></CardId>
                <CardName>visa</CardName>
                <ValidDate>12-10-2004</CardName>
            </Payment>
        </PurchaseOrder>
```

3.3.7 XKMS - Key Management Specification.

A linguagem XML é de fato a linguagem preferida do mercado para efetuar troca de dados confidenciais entre aplicações de negócios na Internet. A segurança para esse tipo de negócio pode ser proporcionada por diversas formas utilizando desde simples senhas de acesso ate certificados digitais para autenticação, sigilo etc. Com o aumento da preocupação em prover mecanismos de segurança eficaz, cresce a demanda de utilização da Infra-estrutura de Chaves Públicas (ICP) nos processos de negócios que utilizam certificados em seus mecanismos de segurança.

As questões relacionadas a mecanismos de segurança como sigilo, autenticidade, não repúdio etc. nas transações eletrônicas efetuadas via mensagens SOAP/XML, foram resolvidas pelos padrões XML Digital Encryption e XML Digital Signature já discutidas neste capitulo em parágrafos anteriores. A questão referente a integração da Infra-estrutura de Chaves Públicas (ICP) com aplicações baseadas em XML (principalmente aquelas utilizando os padrões XML Digital Encryption e XML Digital Signature em seu processos de negócios), estão sendo tratadas pela especificação XML Key Management Specification (XKMS).

A integração com infra-estrutura de ICP é extremamente essencial para ajudar no processo de cifrar, descifrar, assinar digitalmente e verificar autenticidade de documentos quando se utiliza XML encryption e XML digital signature. O protocolo XKMS foi concebido justamente com o objetivo de possibilitar que aplicações baseadas em XML integre as diferentes infra-estruturas de ICP conhecidas, Pretty Good Privacy – PGP [PGP06], Simple Public Key Infrastructure – SPKI [SPKI98], e Public Key Infrastructure X.509 – PKIX [PKI06] a mais utilizada. A questão é saber qual dessas infra-estruturas de ICP é a mais indicada para se utilizar ou integrar. [VER04]

Comentário

Certamente existe a preocupação de integrar e proporcionar interoperabilidade com as diferentes infra-estruturas de ICP existentes atualmente. Por exemplo, a troca de informações criptografadas baseada na utilização de diferentes infra-estruturas de ICP utilizada pelo emissor e receptor da informação certamente causará problemas no processo de descriptografia. Isso ocorre em função das sintaxes e semânticas utilizadas pelas diferentes infra-estruturas de ICP existentes. Isto significa que o emissor da informação criptografada deve utilizar a mesmo modelo de infra-estrutura de ICP utilizada pelo receptor, caso contrario haverá erro no processo de descriptografia.

Essa preocupação de integrar e proporcionar interoperabilidade entre diferentes infra-estruturas de ICP faz pleno sentido considerando que não existe nenhuma padronização de utilização definindo qual infra-estrutura de ICP utilizar. Por exemplo, na arquitetura SOA pode ser adotado o modelo de infra-estrutura de ICP que melhor atenda as

necessidades do negócio. Uma vez adotado a o modelo de infra-estrutura a ser utilizado, todos os demais serviços que desejam estabelecer comunicação utilizando recursos de infra-estrutura de ICP devem integrar o mesmo tipo de ICP, caso contrário torna a solução complexa.

Percebemos que o mercado de negócios adotou a infra-estrutura de ICP (PKIX) como sendo o principal mecanismo para prover segurança para processos de negócios efetuados via Internet. Tudo isso acaba facilitando o gerenciamento e utilização do protocolo XML Key Management Specification (XKMS) para os processos de negócio baseados em XML e na arquitetura SOA.

O protocolo XML Key Management Specification (XKMS) que consiste de dois outros protocolos distintos, XML Key Information Service Specification (XKISS) e XML Key Registration Service Specification (XKRSS), foi desenvolvido pela iniciativa formada pelas empresas VeriSign, Microsoft and WebMethods. Os principais objetivos do protocolo XKMS ilustrado na Figura 07 é permitir a integração de aplicações baseadas em XML com a infra-estrutura de chaves públicas (ICP) e reduzir a complexidade de administração e utilização de certificados digitais pelas aplicações XML clientes

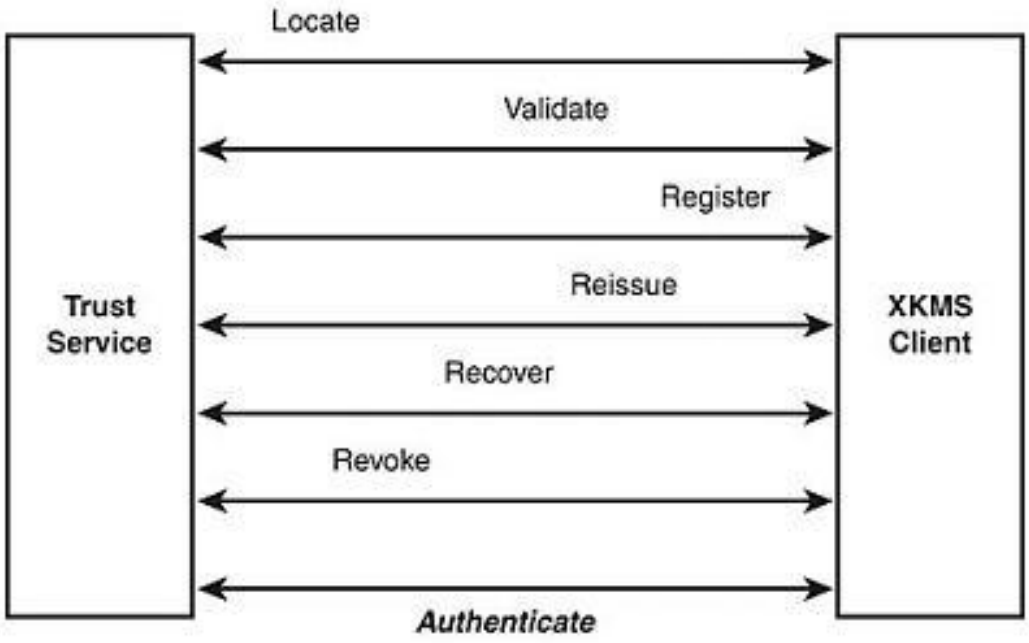


Figura 07 - Tipos de mensagens XKMS e o relacionamento Client/Trust Service, [PUB04].

O XML Key Information Service Specification (XKISS) é o protocolo utilizado pelas aplicações baseados em XML para localizar e validar informações das chaves públicas contidas em documentos XML criptografados ou assinados digitalmente. O protocolo XML Key Registration Service Specification (XKRSS) tem a finalidade de proporcionar os mecanismos para registro, re-emissão, revogação e recuperação da chave pública [SEC03]. Esses dois protocolos XKISS e XKRSS possibilitam a comunicação com o XKMS Server através de mensagens XML trocadas sobre o protocolo Simple Object Acess Protocol (SOAP) [LAT05].

O serviço XKMS proporciona facilidades para os desenvolvedores de aplicações XML que necessitam integrar em suas aplicações os recursos oferecidos pela infra-estrutura de ICP. Isso significa que o provedor de serviço XKMS basicamente estará atuando como uma solução de ICP sem a complexidade e o custo de gerenciamento proporcionado por ela. Essas facilidades vão proporcionar que serviço XKMS se transforme na plataforma segura para os mecanismos de segurança baseados principalmente em XML digital signature e XML encription.

Antes da existência da Standard XKMS, o mercado contava com algumas aplicações e soluções proprietárias para tratar desta questão de integração de aplicações XML com infra-estrutura ICP. Aplicações e soluções proprietárias quase em sua maioria não são compatíveis com outras soluções o que tornava ainda mais grave o problema de interoperabilidade entre essas aplicações. Para resolver esse problema a Standard XKMS foi desenvolvida no conceito “Open Standard” para facilitar e padronizar a integração e interoperabilidade com a infra-estrutura ICP. A lista 07 abaixo descreve as principais características do XKMS.

Lista 07 – Principais Características XKMS, [TIT06].

- XKMS cria uma camada abstrata entre a aplicação e o provedor de ICP facilitando a utilização e absorção da infra-estrutura ICP pela aplicação sem a necessidade de eventuais modificações da mesma.
- XKMS desonera aplicaçoes de conter conhecimentos detalhados de sintaxes e semânticas da infra-estrutura de ICP, permitindo apenas o uso simples de mensagens XML para conversar com o provedor de serviço XMKS.

- XKMS remove a complexidade da aplicação cliente proporcionando aplicações menores e enxutas. Com isso, é possível a utilização dessas aplicações em dispositivos moveis e plataformas de computadores não convencionais.
- A utilização de XKMS também possibilita que aplicações sejam independentes de plataformas e fornecedores o que facilita na portabilidade.

Com a pública ção da Standard XKMS 2.0 pela W3C, novos desenvolvimentos, aplicações e soluções baseadas em XML podem, por exemplo, integrar mecanismos de autenticação, assinatura digital, serviços de criptografia, checar status de revogação de certificados digitais etc.. Tudo isso sem a necessidade de soluções proprietárias para integração com infra-estrutura de ICP. Utilizando o serviço XKMS, as funcionalidades como de relação de confiança estarão armazenados em servidores facilmente acessados via mensagens SOAP/ XML [TRU05].

A Figura 08 ilustra XKMS server proporcionando serviços para registro, localização e validação de chaves públicas. O XKMS server pode ser um provedor de serviço público ou provedor de serviço privado dependendo do modelo de negócio adotado. A utilização do XKMS server como servidor publico faz mais sentido no contexto de negócios via internet, em função disso ressalta outra preocupação com a segurança do próprio provedor de serviço XKMS: Ataques de replay, Denial of Service (DoS), proteção das chaves privadas e públicas etc.

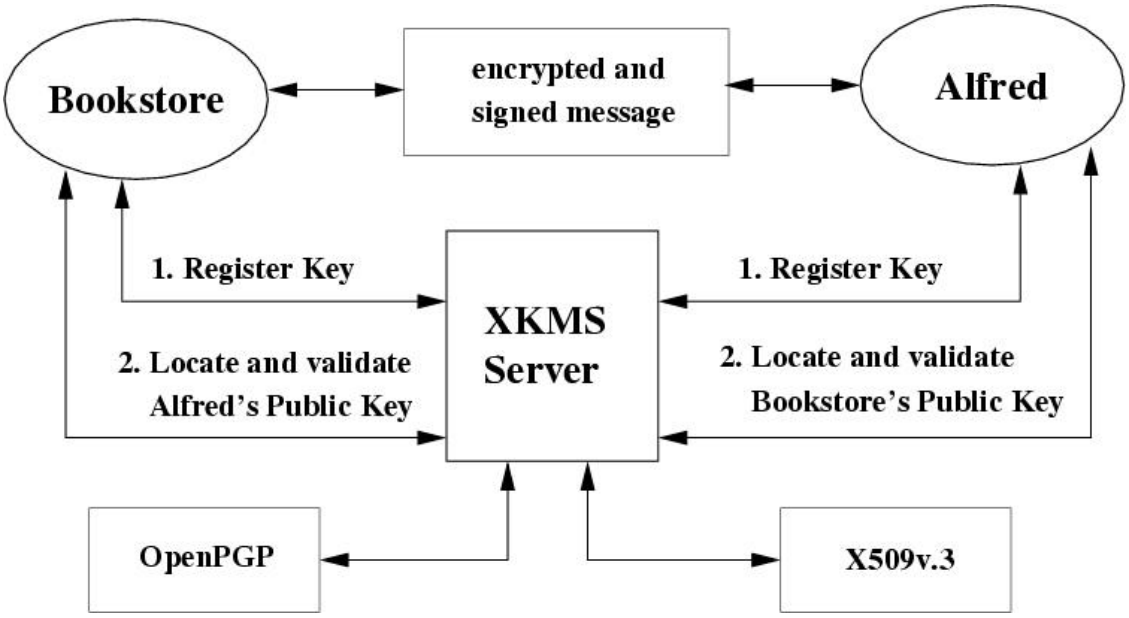


Figura 08 - Registering and using a public key, [HAA04]

Comentário

Se observarmos a Figura 08, percebemos que o XMKS server é a peça chave desse quebra cabeça de relação de confiança proporcionado pela utilização de certificados digitais. De um lado temos um cliente que possivelmente esta efetuando uma compra em uma livraria qualquer, e analisando o desenho observamos também que existem transações criptografadas e assinadas digitalmente no processo de negócio.

Para que o processo de negócio se concretize ambas as partes efetuam a checagem da chave pública no servidor de XKMS para fins de validação de autenticidade das chaves e possível relação de confiança. Uma vez que ambas as chaves são validadas pelo servidor XKMS, o processo de negócio pode então concretizar a transação. Todo esse processo foi possível em função do protocolo XKRSS que proporcionou que ambos (cliente e livraria) registrassem suas chaves públicas no servidor XKMS.

Um fator muito importante nesse processo de registro e validação de chaves públicas via servidor XKMS, é o desafio de manter e gerenciar todas as possíveis chaves públicas das autoridades certificadoras que podem estar assinando chaves públicas no

mercado. A localização da chave pública de nada me adianta se eu não confiar na chave pública da autoridade certificadora que assinou a chave pública em questão.

Utilizar um servidor XKMS privado para gerenciamento de chaves públicas para um processo de negócio de leilão como foi exemplificado anteriormente no capítulo 3.2.1, seria um desafio enorme se considerarmos: A possibilidade de existência de diferentes autoridades certificadoras envolvidas nos processos provenientes de diferentes países embasadas em diferentes normas e regras que regem as leis de certificação digital de cada pais.

O processo manual para gerenciamento de certificados de CAs se torna inviável e custoso para situações onde a relação de confiança deve acontecer dinamicamente como exige o nosso exemplo de leilão descrito no capitulo 3.2.1. O processo para confiar nas chaves públicas das autoridades certificadoras raízes será um desafio para os implementadores da solução de XKMS. Tornar essa relação de confiança dinâmica será o maior desafio para o serviço de gerenciamento de chaves públicas, além dela ainda não disponibilizar nenhuma infra-estrutura pública a nivel global certamente não será sincronizada como serviços de DNS .

3.3.7.1 Desafios, segurança e relação de confiança XKMS.

Desafios, segurança e relação de confiança foram apontados em quase todos os tópicos dessa dissertação. Consequentemente em função disso, interoperabilidade entre os serviços foi diagnosticado como sendo um desafio importante a ser considerado. A segurança e relação de confiança existente e proporcionado pelo serviço XMKS em sua eventual adoção para gerenciamento de chaves públicas para um modelo de negócio baseado na arquitetura SOA, pode proporcionar desafios de interoperabilidades.

Algumas considerações deverão ser feitas para que entendamos os desafios de interoperabilidades que o serviço XKMS vai proporcionar em uma implementação. Por exemplo, devemos observar que dentro dos conceitos de infra-estrutura de ICP, geralmente armazenamos

as chaves públicas raízes das CAs off-line para evitarmos qualquer risco de quebra de segurança, isso também se aplica para as chaves privadas que são ainda mais importantes dependendo do contexto do negócio.

Contrariando algumas premissas básicas com segurança o serviço XKMS vai manter on-line as chaves públicas (CAs raízes, chave pública de clientes e fornecedores etc.). Além disso, o processo de registro e validação de chaves públicas proporcionado pelo serviço XKMS, deve conter mecanismo de relação de confiança. Isso quer dizer os serviços XKISS e XKRSS deve assinar digitalmente as suas respostas, e significa que as chaves privadas devem estar disponíveis implicando no armazenamento on-line no servidor XKMS. Outro fator importante e desafiador envolvendo os serviços XKISS e XKRSS é a distribuição das chaves públicas desses serviços.

Evitar ou tentar minimizar os riscos com segurança relacionados ao serviço XKMS não é tarefa fácil. Algumas hipóteses podem ser consideradas desde que observadas as implicações. Evitando a utilização da assinatura digital nas respostas dos serviços XKISS e XKRSS resolveria a questão de armazenar chave privadas dos serviços no servidor XKMS, porém não teríamos como confiar na integridade das mensagens de respostas. Utilizar mecanismos como HSM para processar assinatura digital das mensagens respostas dos serviços seria uma opção, no entanto esse método poderia causar uma dependência de plataforma e tecnologia contrariando os conceitos da arquitetura SOA e dificultando a interoperabilidade entre serviços [BXSA06].

Outras considerações de segurança também devem ser consideradas em um modelo de negócios envolvendo relação de confiança através de XKMS services. Sabemos que XKMS services ainda não tem uma abrangência global, e que provavelmente as grandes empresas vão optar por gerenciar seus próprios serviços XKMS para proporcionar gerenciamento das chaves públicas de seus possíveis parceiros de negócios. Em situações como essas, a relação de confiança entre serviços corre risco de integridade caso a infra-estrutura de XKMS for hospedado, por exemplo, por uma empresa de hospedeira de servidores. Dependendo do tipo de negócio em questão, problemas contratuais poderão levantar questionamento com relação a legalidade de terceirizar processos de negócios que envolvem relação de confiança e assinatura digital.

3.3.7.2 XML Key Information Service Specification (XKISS).

O XML Key Information Service Specification (XKISS) é o protocolo utilizado pelos serviços baseados em XML para localizar e validar informações das chaves públicas contidas em documentos XML encriptados ou assinados digitalmente.

Por exemplo, quando se recebe um documento XML assinado digitalmente, o elemento <KeyInfo> da assinatura digital especifica o metodo de recuperação do certificado X.509. Em função da complexidade e dificuldade da aplicação XML cliente de resolver o URL (caminho opcional para encontrar informações sobre a chave pública) informado ou processar o certificado digital para extrair parâmetros da chave pública, a tarefa é então confiada ao servidor de serviços XKMS [PB06]. A Figura 09 abaixo ilustra o processo de localização e resolução de nome.

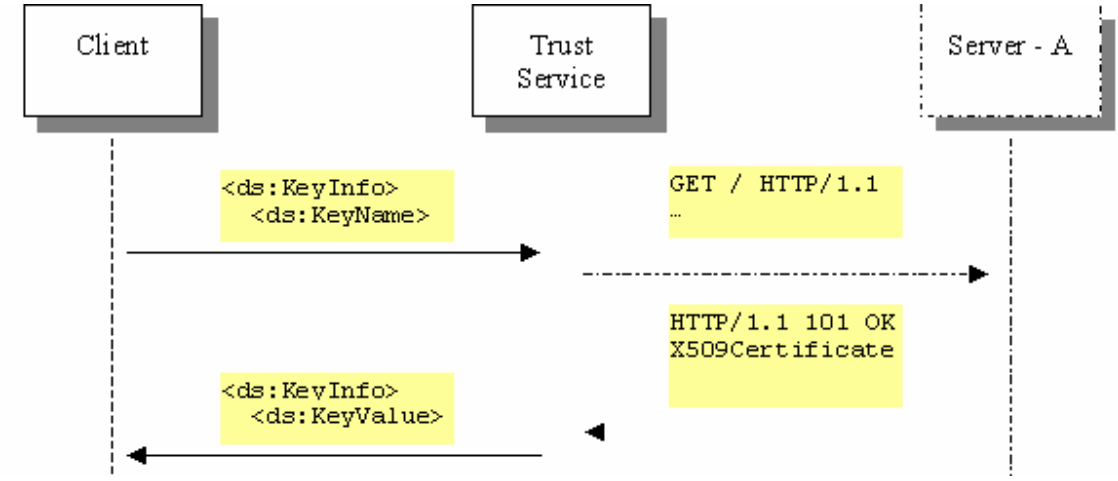


Figura 09 – Serviço de localização – Resolução de Nome, [XKMS05]

Uma vez obtido a informação do serviço de localização, o serviço de validação poderá verificar a validade da assinatura digital. O serviço de validação passa a responsabilidade de verificação e checagem de certificados revogados para o provedor de serviços XKMS. A aplicação cliente desta forma não necessita conhecer a complexidade que envolve o processo de validação e checagens necessárias. A única coisa que a aplicação necessita conhecer é o resultado da solicitação de validação da informação requerida, ou seja, se a chave pública em questão é valida ou não [PB06]. A Figura 10 abaixo ilustra o processo de validação.

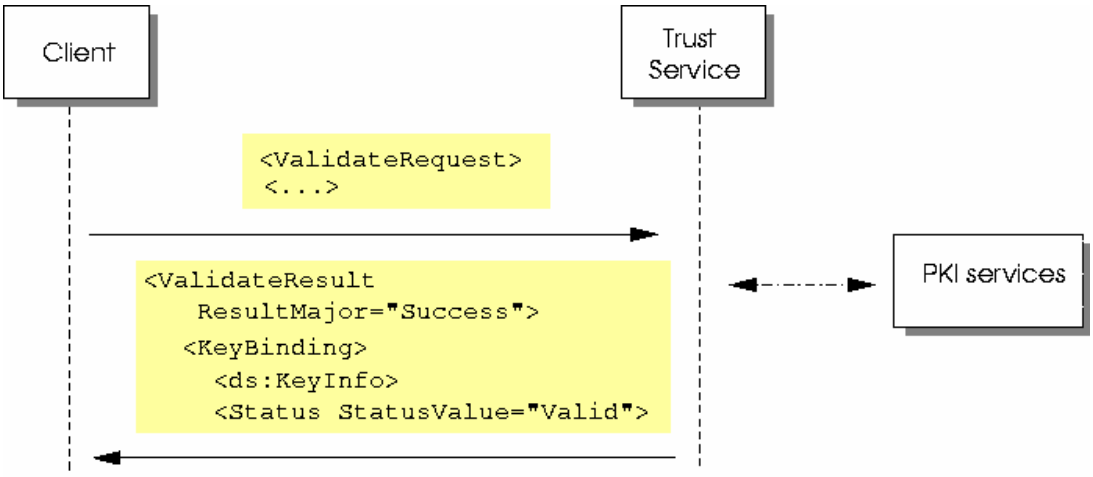


Figura 10 – Serviço de validação, [XKMS05]

3.3.7.3 XML Key Registration Service Specification (XKRSS).

O XML Key Registration Service Specification (XKRSS) tem a finalidade de proporcionar os mecanismos para registro da chave pública, re-emissão, revogação e recuperação. A Figura 11 abaixo ilustra uma infra-estrutrua de serviços XKMS publico acessível via internet.

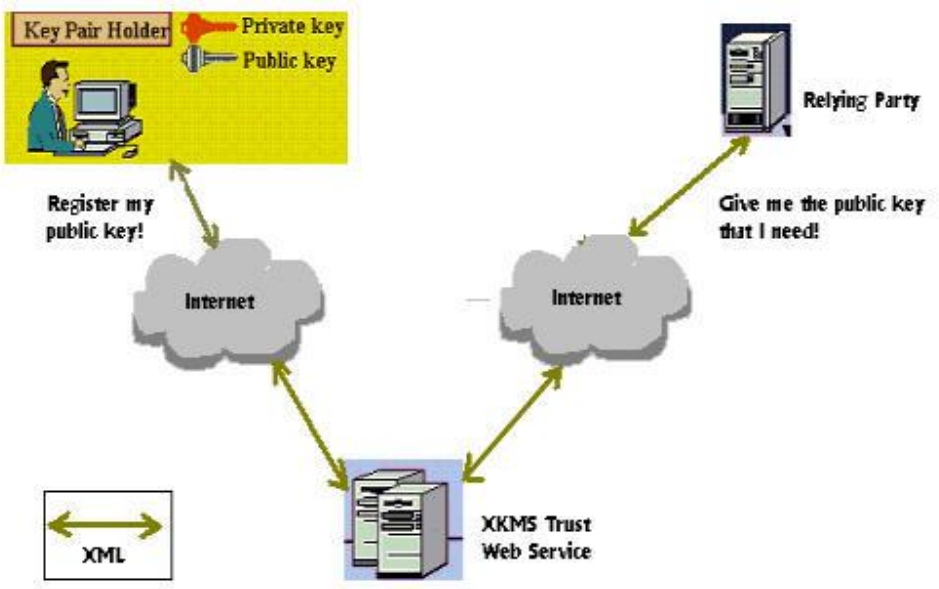


Figura 11 – XKMS Trust Web Service, [TRUS05]

O registro das chaves pode ocorrer de duas maneiras, por exemplo, o cliente gera o par de chaves e envia a chave pública com outras informações (como nome) para o provedor de serviços XKMS para registro, ou o próprio servidor de serviços XKMS gera o par de chaves para o cliente registrando a chave pública e enviado o chave privada para o cliente posteriormente. Neste caso o servidor de serviços XKMS tem a opção de armazenar uma copia da chave privada do cliente mediante solicitação para backup de segurança. E o processo de envio da chave privada para o cliente deve ser através de conexão segura.

Comentário

Observamos que o serviço XKRSS tem a funcionalidade de gerar o par de chaves (pública e privada) e consequentemente armazenar uma copia da chave privada para efeito de backup. Essa opção de armazenar uma copia da chave privada no próprio servidor de XKMS não seria uma boa pratica de segurança. Como já observamos anteriormente, o armazenamento da chave privada em um ambiente que esta constantemente on-line e provavelmente acessível via internet pode colocar em risco a integridade do serviço. Mesmo porque para se ter a habilidade de gerar um par de chaves (pública e privada), o serviço XKMS deve conter em seu poder uma chave privada também, e como já mencionamos armazernar chaves privadas em um servidor on-line não seria uma boa pratica de segurança.

O pedido de renovação ocorre para chaves já armazenadas no provedor de serviços XKMS. A renovação é muito similar ao processo inicial de registro da chave pública. O processo de renovação também ocorre para certificados que foram gerados por uma infra-estrutura de chaves públicas, esse processo pode ser automático e não requer a interação da aplicação cliente.

O serviço de revogação permite que clientes revoguem registros efetuados anteriormente por motivos diversos, exemplo, erro de informação contida na chave pública, problemas de quebra de sigilo da chave privada, cancelamento da chave privada por outros motivos etc.

O serviço de recuperação possibilita a recuperação de chaves privadas que foram perdidas por algum motivo. No entanto esse serviço só é possível se o par de chaves foi criado pelo provedor de serviços XKMS e a chave privada foi mantida armazenada no servidor XKMS para efeito de backup em situações como essa.

Comentário.

O protocolo XML Key Management Specification (XKMS) foi criado justamente para resolver as questões de integração com infra-estrutura de chaves públicas como foi observado nos parágrafos anteriores. As vantagens para aplicações clientes baseadas em XML foram inúmeras. Por exemplo, houve uma redução da complexidade em utilizar a infra-estrutura de chave pública ICP, ocorreu o aumento da portabilidade das aplicações baseadas em XML por se tornarem independente de plataforma e fornecedor etc. Além disso o XKMS é uma especificação Open Standard que foi largamente adotada pelas empresas de software e hardware.

As desvantagens também estão presentes nos serviços proporcionados XKMS como em qualquer outra aplicação. A complexidade, por exemplo, que antes era absorvida pela aplicação XML cliente agora foi transferida para o serviço XKMS. Outros fatores como, gerenciamento centralizado do serviço XKMS, a exposição do serviço XKMS na DMZ [YEU06], a falta de provedores públicos de serviços de XKMS, a possibilidade de cobrança exagerada pelo uso de serviços XKMS etc. são algumas das desvantagens que podemos antecipar somados aos riscos de segurança do próprio serviço identificados anteriormente. Todas essas desvantagens mencionadas se tornam desafios para as empresas de hardware e software que apostaram na demanda do mercado impulsionada pela arquitetura SOA.

Os administradores de serviço XKMS assumem a tarefa de manter a infra-estrutura de chaves públicas proporcionado pelo serviço XKMS integra e segura. Essa tarefa é complexa para empresas que optarem em prover serviços de XKMS para seu domínio interno. A complexidade e custo de gerenciamento pode ser um problema se o serviço de XKMS for oferecido para parceiros de negócios via internet.

O fato que leva a essa complexidade e custo de gerenciamento para empresas que optarem em manter um provedor de serviços XKMS, é a infra-estrutura requerida. Manter um provedor de serviço XKMS íntegro e seguro pode se tornar muito caro. É necessário considerar a alta disponibilidade do provedor de serviço, a segurança física do servidor, as pessoas capacitadas para implementação e manutenção, infra-estrutura de DMZ para prestação de serviços via internet, mecanismos de contingência, backup, procedimentos de segurança bem elaborados etc.

O serviço XKMS tem a característica de centralizar o gerenciamento de chaves em um único servidor possibilitando vários tipos de desvantagens. A desvantagem mais relevante são os ataques DoS/DDoS (Denial of Service or Distributed Denial of Service), frequentemente utilizados contra servidores de serviços críticos que oferecem serviços centralizados na Internet como este. A probabilidade de falha na infra-estrutura XKMS centralizada e alta, no entanto existem mecanismos de segurança para minimizar e contornar essas possíveis falhas como DoS/DDoS, por exemplo, XML firewall, mecanismos que certamente que somarão custos elevados a infra-estutura de XKMS.

Além da possibilidade de um ataque de DoS/DDoS contra o servidor de serviços XKMS em função da sua exposição a internet, existe também a preocupação com a segurança física das chaves privadas dos serviços que poderão estar armazenadas no servidor como já foi observado anteriormente. Para reverter essa situação se faz necessário a utilização de servidor de XML proxy [XTR06] para evitar acesso direto ao servidor de serviço XKMS e talvez solução de HSM (Hardware Security Module) para proteção da chave privada das interfaces, clientes etc., observando todas as implicações mencionadas anteriormente. Esses também são mecanismos de segurança que somarão custos elevados de gerenciamento e manutenção a infra-estrutura de XKMS.

Caso as empresas optarem em não utilizar os serviços de XKMS privados em função da complexidade de manter a infra-estrutura, do alto custo de gerenciamento e manutenção, elas podem optar pela utilização de serviços públicos. O desafio vai ser

encontrar uma infra-estrutura de XKMS pública disponível globalmente e pronta para ser utilizada. A empresa Verisign que foi uma das empresas que ajudaram a desenvolver o serviço XKMS, é uma das poucas empresas que mencionam obter essa infra-estrutura disponível para o mercado, embora não existe referências de casos de sucessos em sua página na web.

Além do problema de não haver ainda muitas empresas oferecendo serviços de XKMS, existe o risco do valor cobrado pela utilização desses serviços serem altos. Atualmente, é extremamente custoso manter uma infra-estrutura de PKI gerenciada através de empresas prestadoras desses serviços, como é o caso do serviço denominado "PKI gerenciada" oferecido pela empresa Verisign. O custo desta infra-estrutura de "PKI gerenciada" tende a ficar maior dependendo do número de certificados digitais que se utiliza.

Os serviços de XKMS que serão oferecidos por empresas como Verisign, certamente estarão praticando preços semelhantes aos preços já praticados pelos serviços de "PKI gerenciada". Podemos chegar a essa conclusão pelo fato que a infra-estrutura utilizada para manter os serviços de XKMS é praticamente a mesma utilizada para manter atualmente os serviços de "PKI gerenciada". Isso certamente vai de certa forma fazer com que as empresas recuem em adotar esses serviços públicos para implementarem serviços de XKMS privados.

A adoção da infra-estrutura de XKMS privada associada com a provável falta de conhecimento técnico do implementador e da imperativa busca pela redução dos custos com infra-estrutura, pode comprometer a segurança do negócio e de toda infra-estrutura. Em função de todos esses desafios e complexidades, a arquitetura SOA que seria o principal motor propulsor dos mecanismos de segurança para aplicações baseadas em XML, pode simplesmente deixar de ser atraente para o mercado justamente em função da complexidade da infra-estrutura de segurança e desafios de implementação.

3.4 Mecanismos de identificação, autenticação e autorização.

Os mecanismos de identificação, autenticação e autorização sempre foram assuntos que causavam preocupações e situações de desafios para qualquer tipo de implementação de sistema. Essas questões ainda persistem e tendem a aumentar com o fator globalização e com o surgimento de novas arquiteturas como a arquitetura SOA. Por exemplo, um modelo de negócio baseado na arquitetura SOA pode proporcionar através das interfaces acessos a diferentes sistemas de informações, consequentemente mecanismos de identificação, autenticação e autorização deverão estar presente para possibilitar a interoperabilidade do negócio e garantir segurança.

Atualmente as empresas buscam por gerenciadores de identidades que proporcionem funcionalidades importantes como Single sign-on, autorização, interoperabilidade entre aplicações/protocolos, plataformas heterogêneas etc., para minimizar os custos com gerenciamento e possibilitar um melhor controle de acesso nas redes enterprises, arquiteturas EAI, serviços SOA etc. O mercado impulsionado pela promissora demanda por gerenciamento de identidade oferece inúmeras aplicações destinadas a cada tipo de arquitetura e negócio, Enterprise Sign sign-on, Web Sigle Sign-on, Federated Single sign-on etc.

O objetivo dessas inúmeras aplicações mencionadas acima, é eliminar a necessidade dos usuários e serviços efetuarem múltiplas autenticações ocasionadas perante acesso a múltiplos sistemas com diferentes métodos de autenticação para cada uma delas. O conceito SSO utilizado por essas aplicações possibilita que usuários e serviços eliminem o processo de solicitação de login e senha aceitando no lugar tokens, cookies, certificados etc. como forma de identificação, autenticação e autorização do usuário e serviço. Para o usuário ou serviço o processo de SSO é transparente, e ele não de da conta que o processo de SSO esta garantindo acesso a diferentes sistemas.

Diante da existência de inúmeras aplicações destinadas a prover SSO, é interessante observar que o principal objetivo de cada um desses diferentes tipos de aplicações é proporcionar um único login de acesso para todos os sistemas envolvidos, e minimizar o custo de

gerenciamento com a utilização descentralizada de diferentes mecanismos de acessos. Enterprise Sign sign-on, por exemplo, é voltado para domínio local, Web Sigle Sign-on e Federated Single sign-on são voltados para ambientes com múltiplos domínios participantes em ambientes Extranets e Internets respectivamente, certamente a arquitetura SOA através de seus serviços poderá se beneficiar destas aplicações.

Grandes empresas almejam em obter um gerenciador de identidades que proporcione acesso único via SSO e esteja em conformidade com os pré-requisitos importantes como, interoperabilidade entre sistemas heterogêneos, identificação, autenticação, autorização, padrões abertos etc. Com o SSO as empresas se beneficiam com o aumento real da segurança no processo de identificação e autenticação, lucra através do gerenciamento minimizado de login e senha, reduz os serviços de suporte relacionados com autorizações etc.

3.4.1 Gerenciamento de identidade para SOA.

Na arquitetura SOA a preocupação com gerenciamento de identidade que possibilite mecanismos de identificação, autenticação e autorização é a mesma de qualquer outra solução. A arquitetura SOA como já constatamos em capítulos anteriores, tem o propósito de maximizar a utilização da infra-estrutura heterogênea já existente, racionalizar o desenvolvimento de novos códigos e proporcionar uma arquitetura open Standard. Talvez o modelo de negócio proporcionado pela arquitetura SOA pode causar um diferencial nos desafios de implementação desses gerenciadores de identidades.

Inúmeros padrões, especificações surgiram e ajudam a endereçar alguns desafios de implementação relacionados a identificação, autenticação e autorização para arquitetura SOA. A arquitetura SOA tem característica de proporcionar inter-conectividade dinâmica entre serviços que poderão pertencer a um único domínio ou vários domínios, empresas e diferentes países. Com esse grande leque de possibilidades de comunicação entre serviços proporcionados pela arquitetura SOA, faz surgir as preocupações com os desafios de interoperabilidade dos mecanismos de gerenciamento de identidade, interoperabilidade, cumprimento da lei de privacidade de cada pais etc.

Atualmente observamos que a grande maioria das empresas não utilizam gerenciadores de identidades em suas aplicações dentro dos seus próprios domínios. Talvez o grande desafio encontrado por estas empresas seja a interoperabilidade entre sistemas heterogêneos existentes e a complexidade envolvida na implementação de gerenciadores de identidades. Na arquitetura SOA os desafios poderão ser ainda maiores com o crescente estimativa do mercado em implementar a arquitetura SOA para prover Web services. Dependendo do tipo de negócio existirá a necessidade de múltiplos acessos por diversos sistemas heterogêneos ao longo de uma comunicação entre serviços SOA, neste caso a identificação, autenticação e acima de tudo a autorização devem ser garantidas e respeitando os requisitos de segurança.

Negócios on-lines através de Web Services via arquitetura SOA vai demandar uma estratégia bem elaborada para garantir a segurança e principalmente autorização de execução dos serviços. A aplicaçoes que oferecem gerenciamento de identidades e SSO será sem dúvidas o caminho a ser adotado pelas empresas para proporcionar um dinamismo nos processos de negócios sem colocar em risco a segurança. O desafio, no entanto, será encontrar uma maneira de fazer o mecanismo de gerenciamento de SSO não falhar por situações de interoperabilidade.

Para negócio baseados na arquitetura SOA pelo menos esses desafios com relação a gerenciamento de identidade poderão deixar de ser desafios no longo prazo. Com a utilização de Padrões abertos tipo XML base da arquitetura SOA, juntamente com os novos padrões SAML, Liberty Alliance e WS-Federation, a questão SSO e Identidade Federada (Federated Identity, como é conhecida) em escala global poderá se tornar realidade. O mecanismo de Federated Identity pode proporcionar para as empresas vários benefícios como:

- Melhoria no login e funções de auditoria.
- Reduzir significativamente custos associados com gerenciamento de senhas.
- Acesso seguro para aplicações heterogêneas existente.

Fonte: [OSFIM06]

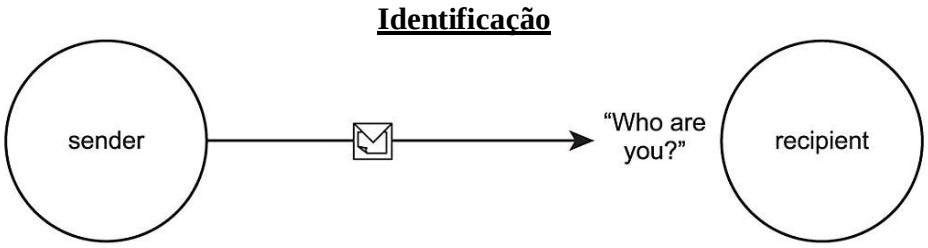
Logo abaixo segue também 3 ilustrações que demonstram os pontos importantes de relação de confiança entre serviços: **Identificação** (Who are you?), **autenticação** (How do I know you are who you say you are) e **autorização** (What are you allowed to do?). Os desenhos

foram retirados do livro de Thomas Erl: Service-Oriented Architecture, Concepts, and Design – pg 259.

O mecanismo de identificação pode se alcançado por diversos métodos de passagem de informação ao sistema. O metodo de identificação através de usuário e senha sem duvidas é o método mais comum utilizado atualmente. Algumas considerações são importantes apontar neste método de identificação, por exemplo, a facilidade de captura de usuário e senha por pessoas mal intencionadas, a possibilidade de ganho de acesso ao sistema através de tentativa e erro etc.

Uma das alternativas para evitar a facilidade de captura de usuário e senha e ganho de acesso ao sistema por tentativa e erro é utilizando o método de autenticação por certificado digital. Atualmente a grande maioria dos sistemas já aceita identificação do usuário via certificado digital. Para sistemas que ainda não disponibilizaram essa funcionalidade, ainda existe a opção (depende da aplicação) de utilizar aplicações que extraem as credenciais do certificado digital para efetuação do login do ambiente desejado. Dessa forma a utilização do certificado digital é possível, porém existirá um sistema especializado para efetuar a extração das informações necessárias para identificação no sistema.

Utilizar mecanismo de identificação através de usuário e senha para uma aplicação baseada na arquitetura SOA não faz muito sentido. Os processos de negócios baseados na arquitetura SOA poderão acessar diferentes tipos de serviços. Geralmente, para obter acesso ao serviço desejado é necessário se identificar, provar autenticidade para obter autorização de acesso pertinente ao seu negócio. Em resumo, mecanismo de autenticação por usuário e senha inviabilizaria a solução, causaria custo elevado de gerenciamento e proporcionaria um alto risco a segurança do negócio, ou seja, uma melhor opção para mecanismo de identificação para negócios baseados em SOA seria a utilização de certificado digital.



An identity is a claim made regarding the origin of a message.

Figura 12 – Identificação, [ERL05]

O mecanismo de autenticação complementa o mecanismo de identificação no sistema. Considerando o mecanismo de identificação por usuário e senha, a autenticação para esse mecanismo seria a própria senha. O usuário fornece o usuário de acesso ao sistema e na seqüência informa a senha de acesso que prova que ele é a pessoa que detêm a informação senha de acesso. Foi observado anteriormente que esse método de identificação e autenticação não é muito seguro, portanto não é recomendável a utilização em ambientes como Internet etc.

Para mecanismos de identificação utilizando certificado digital, o processo de autenticação é todo baseado na infra-estrutura de ICP através na relação de confiança firmada entra as partes envolvidas em um negócio. Esse método de identificação e autenticação através de certificado digital, vem se mostrando muito eficaz e seguro para ambientes inseguros como Internet. Processos de negócios baseados na arquitetura SOA estão adotando a certificação digital para garantir a segurança nos mecanismos de acessos as interfaces, criptografia dos dados e assinatura digital das informações mais relevantes conforme requisitos do negócio.

Algumas situações com relação a utilização de certificado digitais devem ser consideradas. A possibilidade de surgimento de problemas com interoperabilidades pode ocorrer com a utilização de certificados digitais caso não seja observada algumas premissas básicas, como por exemplo, estabelecimento de relação de confiança entre as interfaces antecipadamente. Teoricamente, resolvendo problemas com relação de confiança possibilitaria acesso sem maiores complicações, porém para atingir uma relação de confiança plena entre os ambientes é necessário um esforço considerável. Como foi discutido anteriormente neste capítulo, garantir interoperabilidade dos serviços pode proporcionar inúmeros desafios para os implementadores.

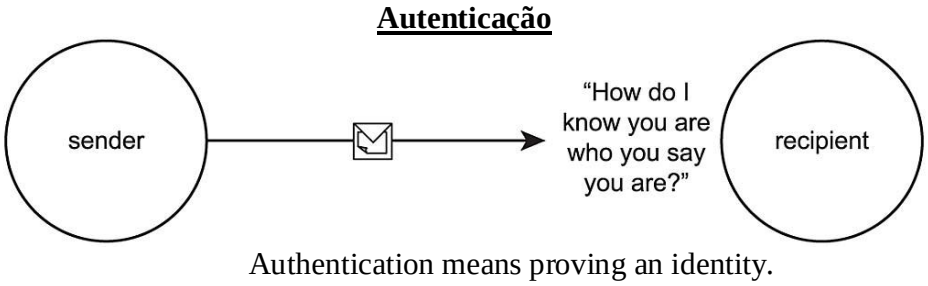


Figura 13 – Autenticação, [ERL05]

Autorização é o resultado do processo das fases anteriores composta pelos mecanismos de identificação e autenticação. Após o usuário ou serviço passar as credenciais de identificação e autenticação seja ela através de usuário e senha ou certificado digital, ele recebe autorização de acesso respeitando o seu perfil de usuário e serviço. A autorização de acesso juntamente com as atividades que o usuário ou serviço pode fazer, deve acompanha-lo durante sua navegação nos sistemas. Manter as autorizações adquiridas no primeiro acesso e leva-las para outros ambientes acessados via SSO pode ser um desafio de implementação.

A arquitetura SOA sem duvidas vai se beneficiar através dessa característica de manter as autorizações adquiridas quando acessar outras interfaces. Certos casos irão demandar um dês-incremento ou ate mesmo um incremento das autorizações adquiridas anteriormente, isso é algo que os requisitos de segurança de cada negócio determina.

Figura 14 – Autorização, [ERL05]

Atualmente o mercado tem a opção de adotar uma dos 3 padrões disponíveis para gerenciamento de identidade, SAML - Security Assertion Markup Language, Liberty Alliance e WS-Federation. O provável é que soluções voltadas especificamente a gerenciamento de identidades (Identity federated), irão prover suporte e compatibilidade entre os padrões disponíveis. O consumidor final por sua vez terá que entender qual Standard melhor atende os requisitos da empresa em uma implementação SOA, antes de optar pela solução de gerenciamento de identidade para sua empresa.

Security Assertion Markup Language (SAML) foi a primeira Standard criada para habilitar gerenciamento de identidade (Identity Federated). Em 2002 SAML foi aprovado pela organização conhecida como Organization for the Advancement of Structured Information Standards (OASIS).

Liberty Alliance Project é uma aliança de mais de 150 empresas, sem fins lucrativos e governamentais comprometidas em desenvolver a Open Standard para Federated Network Identity. O protocolo Liberty 1.1 (atualmente chamado ID-FF 1.1) foi desenvolvido para endereçar as deficiências reais de uso do SAML 1.0, conhecidos como Global logout etc.

Por último, BEA Systems Inc., IBM, Microsoft Corp., RSA Security Inc. and VeriSign Inc, disponibilizaram em 2003 a especificação chamada Web Services Federation (WS-Federation). WS-Federation juntamente com WS-Trust e WS-Policy, formam uma estrutura para requisição e emissão de token com capacidade de Identity Federation [Atul05].

Em função das confusões existentes entre os termos **Federation e Federated Identity,** é interessante descrever a definição de cada um deles conforme segue.

- **Federation:** O acordo, Standard e tecnologias que faz a identidade e seus atributos portável através domínios autônomos.

- **Federated Identity**: Uma associação entre identidades de diferentes domínios. A "Federation" deve estar em conformidade antes que a Federeted Identidy poça ser implementado.

Fonte: [EdOrt05]

Capítulo 4

4 Considerações Finais.

Embora a arquitetura SOA tem se mostrado muito atrativa para ambientes de negócios, algumas considerações com relação aos aspectos de segurança devem estar esclarecidas. Atualmente observa-se que a grande maioria dos projetos corporativos envolvendo tecnologias como Web Services, EAI entre outros, não consideram os requisitos de segurança como sendo uma fase prioritária do projeto/escopo de implementação. Então é natural que uma eventual migração ou adesão da arquitetura SOA ou novos desenvolvimentos também não irão considerar itens de segurança como fase importante de um projeto de implementação.

Em função do comportamento constatado com relação a falta de planejamento com os requisitos de segurança nos processos de negócios, cuidados importantes devem ser tomadas ao se decidir optar por uma arquitetura SOA. Procurar, por exemplo, entender as diferenças existentes entre as aplicações EAI e arquitetura SOA para casos de migração, e analisar os aspectos de segurança diferenciados utilizados para cada uma delas. Considerar previamente todos os requisitos de segurança do negócio e incluir no projeto de implementação ou migração o planejamento dos mecanismos de segurança. Envolver pessoas capacitadas na tecnologia para avaliar as dificuldades e requisitos da arquitetura etc.

O tipo de negócio proporcionado pela arquitetura SOA pode influenciar significativamente na maneira de implementação que as empresas estão acostumadas a gerenciar. Quanto maior for a abrangência dos processos de negócios nos aspectos de comunicação SOA através de serviços via Internet, maior será a demanda por uma definição bem planejada para identificar os requisitos mínimos de segurança e avaliar os aspectos inerentes à implementação. As fases que compõem um projeto (preparação, desenho conceitual, realização, preparação final e entrada em produção) devem contemplar todos os aspectos de segurança da arquitetura SOA, caso contrário a arquitetura corre sérios riscos do ponto de vista de segurança da informação.

4.1 Conclusões.

O objetivo deste trabalho foi realizar um levantamento das principais características da arquitetura SOA, identificar os requisitos de segurança proporcionados por ela e principalmente avaliar os mecanismos de segurança existentes.

No levantamento das principais características da arquitetura SOA, foi constatado que os conceitos proporcionados pela arquitetura podem e devem causar mudança de paradigma para os novos desenvolvimentos baseados na arquitetura SOA. Características como desacoplamento, interface auto-descritiva, interface endereçável etc., certamente causam influência na maneira de arquitetar e desenvolver software.

Avaliando algumas das principais características da arquitetura SOA, identificamos que novos requisitos de segurança surgiram em função do modelo de negócio proporcionado pela arquitetura. Isso pode ser relevante diante das dificuldades de implementação e interoperabilidade que os novos mecanismos de segurança (XML Digital Signature, XML Encryption) podem proporcionar. Além disso existem pontos de atenção a serem considerados diante, por exemplo, da baixa disponibilidade e abrangência de serviços XKMS oferecidos no mercado e da própria segurança física e lógica desse serviço.

Existem também alguns desafios inerentes ao serviço XKMS. Por exemplo, gerenciamento das chaves públicas de CAs e chaves privadas são melhores efetuadas off-line como rege algumas melhores práticas de segurança. No entanto, o serviço XKMS é uma aplicação que deve estar on-line para prover sua funcionalidade, oferecendo um risco para solução como um todo. Certamente existem alternativas para contornar essa situação (soluções de HSM para gerenciamento das chaves), porém elas devem ser utilizadas com critérios obedecendo as características e conceitos da arquitetura SOA, por exemplo, independente de plataforma.

Concluindo, a arquitetura SOA com suas inovações, vantagens e benefícios vieram realmente para ficar. Contudo, o seu sucesso ou adesão no curto e médio prazo conforme estimativa pode estar comprometida se os desafios relacionados aos mecanismos de segurança não forem considerados. Provavelmente a arquitetura SOA vai entrar no mercado em uma

primeira fase substituindo apenas os processos de negócios proporcionados principalmente por EAI. Desta maneira os requisitos de segurança permanecem os mesmos enquanto o mercado ganha mais experiência com a solução e os provedores de solução aperfeiçoem os mecanismos de segurança.

4.2 Trabalhos Futuros

Em relação a trabalhos futuros são feitas algumas sugestões para continuidade no estudo dos desafios dos aspectos de segurança na adoção Arquitetura Orientada a Serviços.

- **Serviço XKMS Público - Distribuído.**

  Para a arquitetura SOA utilizar de forma mais dinâmica os mecanismos de segurança proporcionados pela infra-estrutura de chaves públicas nos seus processos de negócios, é necessário que serviços XKMS públicos sejam disponibilizados a nivel global e sincronizados entre si como serviços de DNS. Um modelo de serviço XKMS distribuído e controlado similar ao serviço de DNS proporcionaria inúmeros benefícios.

  - Redução de custos com manutenção de Serviços XKMS privados.
  - Redução de risco de falta de disponibilidade do serviço.
  - Possibilidade de outros serviços utilizarem a infra-estrutura, exemplo, serviços SMTP.
  -Acesso principalmente serviços de validação de chaves públicas.
  - Acesso a base de dados atualizada de certificados revogados atualizados.
  - etc.

- **Serviço XKMS Privado - Segurança.**

  Serviço de XKMS privado é a opção do mercado em uma implementação da arquitetura SOA, mesmo porque não existem muitos serviços XKMS públicos sendo

oferecidos atualmente. Optar por manter um serviço de XKMS privado pode proporcionar um custo alto de gerenciamento e até colocar em risco a segurança do negócio. Tudo isso certamente depende do grau de utilização dos mecanismos de segurança nos processos de negócios e pelo comprometimento com os princípios de segurança da infra-estrutura.

- Disponibilidade do serviço XKMS na DMZ.
    - Chaves privadas das interfaces.
    - Chaves privadas dos usuários registros (Opcional).
    - Chaves públicas (CAs) confiadas.
- Alta disponibilidade dos serviços.
- Utilização de HSM para armazenamento da chaves privadas baseado na arquitetura SOA.
- Gerenciamento de Chaves Públicas CAs.
- Utilização de XML Proxy.
- Utilização de XML Firewal

Capítulo 5

Referências Bibliográficas

[ANA05] R. Anantharangachar. Demystifying SOA - Myths About SOA Web Services Architecture, - Sep. 22, 2005 http://websphere.sys-con.com/read/121947.htm

[Atul05] Steps for choosing the right federated identity standard for your company, by Atul Tulshibagwale, Trustgenix – Sep. 30, 2005 http://www.computerworld.com/printthis/2005/0,4814,105019,00.html

[BAS05] Basic Security Profile Version 1.0, by Working Group Draft 2005-08-29 http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html

[BOR05] B. Borges, K. Holley, A. Arsanjani. Delving into Service-Oriented Architecture http://www.developer.com/java/ent/article.php/10933_3409221_1Acessado em 08.07.05

[BXSA06] Building an XKMS Service using ASP.NET, by Sébastien Pouliot – Acessado em 05.03.06 http://pages.infinit.net/ctech/xkms-part2.html

[CAP05] G. Caprio. An Introduction to Service-Oriented Architecture, June 8, 2005 http://www.devx.com/codemag/Article/28254/0/page/3

[CEP03] Ceperez. A Definition of "Services" 2003-12-23 http://www.manageability.org/blog/stuff/what-is-a-web-service/view

[CON05] Constructing Software For Service Oriented Architecture http://www.ebpml.org/csfsoa.ppt Acessado em 08.07.05

[CON06] Configuration of XKMS service http://www.cswl.com/whiteppr/tech/xkms.html Acessado em 21.01.06

[COU05] J. Counihan. XML and Competitive Advantage http://www.netdesk.com/CourseInfo/Articles/Developers/xmladvantage.htm Acessado em 10.10.05

[DEL05] Deliverables from the Basic Profile Working Group http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile Acessado em 07.08.05

[DEV05] Development History, by W3C. http://www.w3.org/XML/hist2002 Acessado em 08.07.05

[EdOrt05] What's New in SOA and Web Services?, *By Ed Ort, October 3, 2005* http://java.sun.com/developer/technicalArticles/WebServices/soa2/WhatsNewArticle.html

[ENC05] XML Encryption Syntax and Processing, by W3C. http://www.w3.org/TR/xmlenc-core/ Acessado em 08.07.05

[ERL05] T. Erl. Services-Oriented Architecture, Concepts, Technology, and Design, Prentice Hall PTR August 2, 2005

[EXT05] Extensible Markup Language (XML) 1.0 (Third Edition), by W3C. http://www.w3.org/TR/2004/REC-xml-20040204/ Acessado em 08.07.05

[FB00] W. Ford, M. S. Baum. Secure Electronic Commerce – Second Edition - Prentice Hall PTR, 2000

[FOOD06] The dangers of single sign-on, by Dan Foody – Acessado em 03.03.06 http://www.soa-zone.com/index.php?/archives/22-The-dangers-of-single-sign-on.html

[GAR05] Gartner Group http://www.gartner.com/ Acessado em 10.10.05

[GRA05] P. Gralla. The ROI of Web services and SOA, 04.12.2005 http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci1078192,00.html

[HAA04] H. Haas. Registering and using a public key, 20 May 2004 http://www.w3.org/2004/Talks/0520-hh-xmlsec/slide13-0.html

[HAR06] How Hardware Security Modules Can Optimize Key Protection and Management http://www.ingrian.com/resources/sol_briefs/hsm_sb Acessado em 09.03.06

[HE03] H. He. What is Service-Oriented Architecture? September 30, 2003 http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html

[IETF06] The Internet Engineering Task Force (IETF) http://www.ietf.org/ Acessado em 21.01.06

[INT06] Internet Protocol, by Cisco http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ip.htm Acessado em 10.03.06

[KAY05] D. Kaye. Web Services Strategies http://www.rds.com/doug/weblogs/webServicesStrategies/2002/11/18.html Acessado em 15.11.05

[KIRK05] Identity federation: Is it time to move now?, By Jeremy Kirk - September 15, 2005 http://www.infoworld.com/article/05/09/15/HNidfederation_1.html

[KOD05] R. R. Kodali.What is service-oriented architecture? - 5/06/2005 http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html

[KOD05] R. R. Kodali. What is service-oriented architecture? 15/06/2005. http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html

[LAT05] Latest SOAP versions, W3C http://www.w3.org/TR/soap/ Acessado em 07.08.05

[LIS05] Lista de Profiles e especificações - OASIS http://docs.oasis-open.org/wss/ Acessado em 07.08.05

[LOE05] L. Loeb. Hack Proofing XML - Chapter 5 – XML Digital Signatures, http://www.syngress.com/book_catalog/224_hack_xml/sample.pdf Acessado em 03.06.05

[McA04] The global challenge, By Neil McAllister - September 03, 2004 http://www.infoworld.com/article/04/09/03/36FEidentityfedglobal_1.html

[MED05] Medida provisória MP 2.200-2 que Institui a Infra-Estrutura de Chaves Públicas Brasileira. http://www.icpbrasil.gov.br/ Acessado em 05.12.05

[MMDV02] J. D. Meier, A. Mackman, M. Dunner, S. Vasireddy. Building Secure ASP.NET

Applications: Authentication, Authorization, and Secure Communication, Microsoft Corporation - November 2002. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnnetsec/html/SecNetch10.asp

[MT05] T. McCall. Gartner Analyzes Hottest Topics of 2005: Open-Source, Voice/Data Convergence, Service-Oriented Architecture, IT Utility and Global Sourcing, June 7, 2005 (http://www.gartner.com/press_releases/asset_128638_11.html

[NAT03] Y. V. Natis. Service-Oriented Architecture Scenario,16 April 2003 http://www.g2r.com/DisplayDocument?doc_cd=114358

[NFE06] Projeto Nota Fiscal Eletrônica (NF-e). http://www.portalfiscal.se.gov.br/WebPortalFiscal/notaFiscalEletronica/justificativas.jsp Acessado em 04.04.06

[NG04] E. T. Nakamura e P. L. Geus. Segurança de Redes – 3ª Edição - Editora: FUTURA – ISBN 85-7413-179-2

[OAS05] OASIS (Organization for the Advancement of Structured Information Standards). http://www.oasis-open.org. Acessado em 07.08.05

[ORG06] Organization for the Advancement of Structured Information Standards" (OASIS). http://www.oasis-open.org/home/index.php Acessado em 21.01.06

[OSFIM06] Open Source Federated Identity Management – Acessado em 05.03.06 http://www.sourceid.org/content/primer

[PB06] Phillip M, H. Baker, Warwick Ford. XML Key Management Specification (XKMS) http://www10.org/cdrom/posters/1129.pdf Acessado em 02.02.06

[PGP06] The OpenPGP Alliance http://www.openpgp.org/ Acessado em 25.03.06

[PKI06] Public-Key Infrastructure (X.509) (pkix) http://www.ietf.org/html.charters/pkix-charter.html Acessado em 04.04.06

[PRO06] SSL/TLS Protocol Overview http://ie.activedomain.org/121.htm Acessado em 15.03.06

[PUB04] S. Publishing. Trust, Access Control, and Rights for Web Services, Part 2 - 2004-10-12 http://www.devshed.com/c/a/Security/Trust-Access-Control-and-Rights-for-Web-Services-Part-2/1/

[RES01] E. Rescoria. SSL and TLS: Designing and Building Secure Systems - Addison-Wesley, 2001

[SAMG06] Secure your SOA, by Ash Parikh, Anthony Sangha, and Murty Gurajada - April 10, 2006 http://www.javaworld.com/javaworld/jw-04-2006/jw-0410-webservices.html

[SEC03] Secure Web services, By Sang Shin – 18.03.2003 http://www.javaworld.com/javaworld/jw-03-2003/jw-0321-wssecurity-tote.html

[SER05] Service Composition http://www.serviceoriented.org/service_composition.html Acessado em 15.12.05

[SID02] B. Siddiqui. Exploring XML Encryption, 01 Mar 2002 http://www-128.ibm.com/developerworks/xml/library/x-encrypt/#code1

[SM05] M. Stevens. Understanding Service-Oriented Architecture, July 8, 2005 http://www.developer.com/services/print.php/10928_2207371_1

[SMA01] E. Simon, P Madsen, C Adams. An Introduction to XML Digital Signatures, August 08, 2001 http://www.xml.com/pub/a/2001/08/08/xmldsig.html

[SPKI98] Y. Wang. SPKI, December 7, 1998 http://users.tkk.fi/~yuwang/pública tions/SPKI/SPKI.html

[SSL05] O protocolo SSL http://wp.netscape.com/eng/ssl3/ssl-toc.html Acessado em 07.10.05

[STE05] M. Stevens. Understanding Service-Oriented Architecture. July 8, 2005 http://www.developer.com/services/print.php/10928_2207371_1

[TIT06] E. Tittel. Finessing PKI with XMKS Makes Trust Portable, 01.26.2006 http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci1161904,00.html?FromTaxonomy=%2Fpr%2F285741

[TLS05] O protocolo TLS http://www.ietf.org/html.charters/tls-charter.html Acessado em 07.10.05

[TRA05] Transport Layer Security http://en.wikipedia.org/wiki/Secure_Sockets_Layer Acessado 01.02.05

[TRU05] The Need for XML Trust Infrastructure, by Verisign. http://www.verisign.com/stellent/groups/public/documents/white_paper/005323.pdf Acessado em 08.10.05

[TRUS05] XKMS Trust Web Service www.cswl.com/whiteppr/ tech/xkms.html Acesso em 02.08.05

[TUL05] A. Tulshibagwale. Steps for choosing the right federated identity standard for your company, - September 30, 2005 http://www.computerworld.com/printthis/2005/0,4814,105019,00.html

[UDD05] UDDI http://www.uddi.org/ Acessado em 07.08.05

[VER04] M. Verma. XML Security: The XML Key Management Specification, 27 Jan 2004 http://www-128.ibm.com/developerworks/xml/library/x-seclay3/

[W3C06] The World Wide Web Consortium (W3C) http://www.w3.org/ Acessado em 21.01.06

[WEB05] Web Services Description Language (WSDL) 1.1, by W3C http://www.w3.org/TR/wsdl Acessado em 07.08.05

[WEBD06] Web Services Definition - http://pt.wikipedia.org/wiki/Web_service - acessado em 02.12.06

[WIL05] L. Wilkes. ROI - The Costs and Benefits of Web Services and Service Oriented Architecture, http://roadmap.cbdiforum.com/reports/roi/ Acessado em 12.12.05

[WSD05] WS-Deliverables http://www.ws-i.org/deliverables/Default.aspx Acessado em 07.08.05

[WSI05] WS-I Releases Basic Security Profile Version 1.0 Working Group Draft. http://xml.coverpages.org/ni2004-05-18-a.html Acessado em 08.08.05

[WSI06] Web Services Interoperability Organization (WS-I) http://www.ws-i.org/ Acessado em 21.01.06

[XKMS05] XML Key Management Specification (XKMS 2.0), by W3C – 28.06.2005

http://www.w3.org/TR/2005/REC-xkms2-20050628/

[XML02] XML-Signature Syntax and Processing, by W3C Recommendation 12 February 2002
http://www.w3.org/TR/xmldsig-core/

[XTR06] Xtradyne's WS-DBC - the XML Proxy for Enterprises
http://www.xtradyne.com/products/ws-dbc/xml-proxy.htm Acessado em 07.03.06

[YEU06] B. Yeung. Introduction to Firewall, 01.02.06
http://www.tns.com/firewalls.asp